

Universitat de Lleida  
Escola Politècnica Superior  
Enginyeria en Informàtica

Sistemes informàtics (Treball de final de carrera)

## **Plataforma d'Exàmens en Xarxa**

Autor: Marc Ribes Novau  
Directora: Magda Valls Marsal  
Juliol 2007



# Índex

<b>Capítol 1</b> Introducció .....	4
<b>Capítol 2</b> Tecnologies emprades .....	6
2.1 XML .....	7
2.1.1 Estructura d'un document XML .....	8
2.1.2 Com transformar documents XML .....	10
2.1.2.1 XSLT .....	10
2.1.2.2 XSL-FO .....	13
2.1.3 APIs de Java per a treballar amb XML .....	17
2.1.3.1 SAX .....	17
2.1.3.2 DOM .....	18
2.1.4 Eines de software per a treballar amb XML .....	18
2.2 Java Web Start .....	19
2.3 SSL .....	20
2.3.1 Creació d'un certificat digital SSL .....	22
<b>Capítol 3</b> Desenvolupament de la plataforma .....	24
3.1 Requeriments inicials .....	25
3.1 Temporització .....	26
3.2 Estudi de mercat .....	26
3.3 Arquitectura de la plataforma .....	28
3.4 Disseny de la plataforma .....	30
3.4.1 Mòdul de creació d'exàmens .....	31
3.4.2 Mòdul d'edició d'alumnes .....	32
3.4.3 Mòdul de desplegament en xarxa .....	33
3.4.4 Mòdul de correcció d'exàmens .....	36
3.4.5 Altres consideracions de disseny .....	37
<b>Capítol 4</b> Manual d'usuari .....	39
<b>Capítol 5</b> Conclusions i futures línies de treball .....	68
5.1 Conclusions .....	68
5.2 Futures línies de treball .....	70
<b>Taula d'il·lustracions</b> .....	71
<b>Glossari de termes</b> .....	72
<b>Bibliografia</b> .....	77

# Capítol 1

## Introducció

En el present document es descriu el projecte “Plataforma d’Exàmens en Xarxa”. La finalitat del projecte és la de desenvolupar una plataforma capaç de dur a terme tot el procés de realització d’un examen, des de la part de creació fins a la d’obtenció de les notes aconseguint una simplificació considerable de tot el procés.

PEX (Plataforma d’Exàmens en Xarxa) s’ha dissenyat d’acord amb totes les tecnologies estàndard de codi obert amb la finalitat d’aconseguir el major grau de flexibilitat, accessibilitat i usabilitat possible.

La plataforma inclou tots els instruments necessaris per crear els exàmens, desplegar-los en xarxa, avaluar-los i arxivar-los, estalviant així una gran quantitat de temps als professors.

Les característiques principals de PEX són:

- Independent de la plataforma.
- Basat en tecnologies de codi obert.



- Varis tipus de qüestions (tipus test, descripció, veritat o fals..).
- Generació de l'examen en format HTML i PDF.
- Permet fer exàmens en xarxa.
- Permet controlar i sincronitzar l'inici i la finalització de l'examen.
- Correcció automàtica dels exàmens.
- Diferents tipus d'informes de resultats.
- Elevat grau de personalització dels exàmens.
- Sistema de seguretat per controlar els usuaris i la integritat de les dades.

Les principals tecnologies que intervenen en el projecte són per una banda Java per la realització de les aplicacions que componen la plataforma, XML com a eina per treballar amb les dades dels exàmens, alumnes, resultats i respostes obtingudes dels processos de creació i manipulació, XSLT i XSL-FO per realitzar les transformacions de les dades a format HTML, CSV o PDF segons convingui, Java Web Start per desplegar l'examen en xarxa i SSL per controlar la seguretat de la plataforma.

En aquesta memòria s'expliquen les principals característiques de la plataforma, així com una introducció a les principals tecnologies que han intervingut al llarg del desenvolupament del projecte. També s'inclou un manual d'usuari per tal de poder utilitzar correctament totes les eines proporcionades.

Un glossari de termes, molt útil degut al gran nombre d'acrònims presents i vocabulari a vegades una mica massa tècnic que poden despistar al lector, i l'obligada bibliografia posen el punt i final a aquesta documentació.

Finalment, donar constància de l'interès personal per aquest tema ja que PEX resultarà ser una eina molt útil per la meua actual tasca laboral com a professor en un institut de secundària. Espero que també pugui resultar una eina útil per a moltes altres persones.

Aquest projecte s'ha utilitzat com a prova pilot pel projecte PID 0821: *“El portafoli com a eina per a tutoritzar els treballs fi de carrera i millorar les competències genèriques dels alumnes de l'Escola Politècnica Superior de la Universitat de Lleida”*.

## Capítol 2

# Tecnologies emprades

L'objectiu d'aquest capítol és el d'introduir les principals tecnologies amb les que ha estat desenvolupat el projecte. En cap cas es pretén explicar de forma rigorosa i complexa cada una de les tecnologies però sí fer factible la seva utilització com a petita guia o referència per a totes aquelles persones que es vulguin endinsar en aquest món, o bé, per afavorir que una persona no iniciada en alguna d'aquestes matèries pugui assolir el grau suficient per a comprendre el projecte desenvolupat.

En cada un dels apartats del capítol es procura donar una explicació general, basada en els conceptes que s'utilitzen en el projecte, de cada una de les tecnologies involucrades, acompanyat d'un exemple senzill d'utilització.

En aquest capítol es tracta XML [16] amb les seves possibles transformacions XSLT [18] i XSL-FO [10], Java Web Start [19] i SSL [1].

## 2.1 XML

XML ( *eXtensible Markup Language* ) no és, com el seu nom pot suggerir, un llenguatge de marca. XML és un metallenguatge que ens permet definir llenguatges de marca adequats a usos determinats. Ha guanyat molta popularitat en els darrers temps degut a que es tracta d'un estàndard obert i lliure. Va ésser creat pel consorci *World Wide Web*, W3C ( els creadors del *www* ), amb la col·laboració dels representants de les principals companyies de software.

La primera especificació de XML va aparèixer el 1998, però ja havia estat proposat dos anys abans, al 1996. Des d'aleshores ha tingut un creixement accelerat, que s'espera que continuï durant els propers anys i actualment ja és una eina imprescindible per a la majoria de programadors.

L'HTML ( *HyperText Markup Language* ) s'ha convertit en el llenguatge estàndard del *World Wide Web*. S'ha confirmat amb un estàndard acceptat i aprovat per la indústria. Tant l'HTML com l'XML es poden considerar com a descendents del SGML ( *Standard Generalized Markup Language* ). Es pot dir que SGML conté els fonaments de XML i HTML. L'inconvenient que presenta SGML, radica en què per ser molt flexible i molt general, realitzar un anàlisi sintàctic d'un document en aquest llenguatge i especificar-ne la seva estructura no és una feina fàcil. XML és més exigent que SGML en la sintaxi, entre altres coses, i això en simplifica el seu processament.

La diferència més important entre XML i HTML és que mentre que en HTML la sintaxi i la semàntica del document estan definides, XML permet definir sintaxi de documents. Els document HTML no estan ben formatats, hi ha etiquetes que no tenen la seva corresponent etiqueta de tancament.

Les principals avantatges que comporta XML són:

- És un estàndard. Ningú el controla.
- Emmagatzema la informació de forma neutra, fet que fa que el mateix contingut pot ser publicat en diferents medis. Portabilitat.
- Es pot modelitzar qualsevol tipus de dades organitzat jeràrquicament.
- Les dades es poden validar estructuralment. Afavoreix la reutilització de codi perquè els *parsers* ja estan fets.

### 2.1.1 Estructura d'un document XML

Els fitxers XML són fitxers de text, que en principi estan en codi unicode, però poden utilitzar altres alfabetes. Existeixen 5 caràcters especials en XML : els símbols menor que (<), major que (>), les cometes dobles ( “), l’apòstrof (‘) i el caràcter (&). Els símbols major que i menor que s'utilitzen per a delimitar els elements que donen estructura al document, les cometes i l’apòstrof contenen valors d'atributs i el caràcter & s'utilitza per representar entitats.

Hi han dos tipus principals de documents en XML:

- Documents Ben Formats. Un document XML ben format ha de seguir dos regles: 1) Tot element ha de tenir una etiqueta d'obertura i una de tancament. 2) Ha d'existir un element arrel que contingui tots els altres elements. Aquest tipus de documents no tenen perquè adherir-se a un *Document Type Definition* (DTD). Un DTD determina els elements estructurals i definicions de marca que es poden utilitzar en el document.
- Documents Vàlids. Un document XML vàlid està definit per un DTD, que conté la gramàtica del document que defineix quins tipus d'elements, atributs i entitats pot contenir el document. El DTD defineix l'ordre i també l'ocurrència dels elements.

Tot document XML comença per la següent línia:

`<?xml version="1.0"?>`

Després opcionalment es pot posar, si és el cas, quin és el DTD del document XML:

`<!DOCTYPE exemple SYSTEM "exemple.dtd">`

El document té estructura d'arbre, la seva arrel és l'element declarat a la marca DOCTYPE.

Cada element està delimitat per una marca d'obertura i una de tancament:

`<element> contingut </element>`

La marca de tancament no es pot ometre, però si l'element està buit pot abreviar-se:

```
<element/>
```

El tenir una estructura d'arbre implica que un element no pot tancar-se si abans no s'han tancat els que conté.

```
<element>
  <elementFill>
</element>
  </elementFill>
```

**Incorrecte**

```
<element>
  <elementFill>
  </elementFill>
</element>
```

**Correcte**

Els elements poden tenir atributs que han d'anar entre cometes, ja siguin dobles o simples.

```
<element tipus="numeric">
```

Es poden posar comentaris, la seva sintaxi serà de la forma:

```
<!--això és un comentari -->
```

Exemple de document XML que conté la informació dels llibres d'una biblioteca:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<biblioteca>
  <llibre>
    <titol>Java and XML</titol>
    <autor>Bret McLaughlin </autor>
    <editorial>O'REILLY </editorial>
    <categoria classe="programació"/>
    <ISBN>0-596-00197-5</ISBN>
    <comentaris>Llibre que explica com treballar amb XML utilitzant
      Java com a llenguatge de programació </comentaris>
  </llibre>
  <llibre>
    <titol>LDAP programming with Java</titol>
    <autor>Rob Weltman</autor>
    <editorial>Addison Wesley</editorial>
    <categoria classe="programació"/>
    <ISBN>0-201-65758-9</ISBN>
    <comentaris>Programació amb Java de serveis LDAP.
      673 pàgines</comentaris>
  </llibre>
  ...
</biblioteca>
```

**Il·lustració 1. Exemple arxiu XML**

### 2.1.2 Com transformar documents XML

Per tal de definir una presentació o format per a un document XML s'utilitza el XSL ( *eXtended Style Language* ). Bàsicament XSL és un llenguatge que defineix una transformació entre un document XML d'entrada i un altre document de sortida XML, HTML o de qualsevol altre format (Postscript, PDF, RTF, etc.). És una especificació desenvolupada dins del W3C per aplicar format als document XML de forma estandaritzada. Els fulls d'estil XSL són documents XML.

Un mateix document XML pot tenir diferents fulls d'estil XSL que permetran mostrar el contingut del document XML en diferents formats ( HTML, PDF, RTF, VRML, ...).

L'aplicació d'un full d'estil XSL a un document XML es pot produir tant en l'origen ( per exemple, un *servlet* que converteixi de XML a HTML per a que pugui ser mostrat en un navegador ), o en el mateix navegador sempre que aquest ho suporti.

El XSL és un llenguatge per escriure fulls d'estil que consta de tres parts:

- XSLT, que és un llenguatge de transformació, sovint útil fora de XSL, per ordenar, afegir o borrar etiquetes i atributs.
- XSL-FO, un llenguatge de formateig, que no és més que un vocabulari XML per especificar objectes de formateig (FO).
- XPath, un llenguatge de baix nivell que permet definir parts d'un document XML.

#### 2.1.2.1 XSLT

XSLT és la part més important de les especificacions XSL. És la part de XSL que és utilitzada per a transformar un document XML en un altre document XML, o algun altre tipus de document que sigui reconegut per un navegador, com poden ser HTML i XHTML. Normalment, XSLT duu a terme aquests canvis transformant cada element XML en un element (X)HTML. Els navegadors Netscape 6 i Internet Explorer 5 no suporten de forma completa la recomanació W3C XSLT. No és fins a la versió 7 del navegador Netscape i la 6 del Internet Explorer que compleixen completament amb la recomanació XSLT del W3C.

XSLT pot afegir nous elements al fitxer de sortida i també pot eliminar-los. També pot prendre decisions sobre quins elements mostrar, ordenar-los segons algun criteri...

## Estructura d'un document XSLT

En un document o full d'estil XSL, l'element arrel és sempre :

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

*xsl-output* controla el tipus de sortida que produirà la fulla XSL:

```
<xsl:output method="xml" indent="yes"
doctype-public="-//W3C//DTD XHTML 1.0 Strict//EN"
doctype-system="http://www.w3.org/TR/xhtml1/DTD/strict.dtd"/>
```

En aquest cas es produiria un altre document XML (en concret XHTML). Si es vol per exemple, obtenir un document HTML s'ha de canviar el valor de l'atribut “*method*” a html.

```
<xsl:output method="html"/>
```

L'element més important és *xsl:template*, que defineix una transformació. Disposa d'un atribut “*match*” que indica els elements als quals s'aplica la transformació. Les etiquetes que contingui aquest element i no pertanyin a l'espai de noms XSL, seran posades tal qual al document de sortida.

*xsl:apply-templates* aplica els *xsl:template* associats a cada element que figuri a la part del document XML que s'estigui processant. Si aquest element no conté atributs, s'aplicaran els patrons (*templates*) a tots els elements que hi hagi en l'ordre en que apareixen al document XML. L'ús de l'atribut “*select*” farà que només es processin els *templates* de certs elements.

```
<xsl:template match="persona">
  <tr> <td>
    <apply-templates select="nom">
  </td> <td>
    <apply-templates select="telefon">
  </td> </tr>
</xsl:template>
```

Altres elements que es poden utilitzar en les fulles d'estils XSL poden ser:

Element	Descripció
<b>xsl:include</b>	Inclou una altra fulla d'estil XSL en el punt en el que apareix
<b>xsl:import</b>	Defineix una fulla d'estil XSL a la qual s'acudirà en cas de que alguna part no estigui definida en la fulla actual
<b>xsl:value-of</b>	Obté el valor d'un node
<b>xsl:variable</b>	Declara una variable
<b>xsl:text</b>	Escriu text de forma literal a la sortida
<b>xsl:if</b>	Es processa si es compleix la condició
<b>xsl:for-each</b>	Itera sobre un conjunt d'elements
<b>xsl:when,</b> <b>xsl:choose i</b> <b>xsl:otherwise</b>	Serveixen per expressar condicions múltiples

Exemple d'un tipus de conversió del fitxer biblioteca.xml a HTML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<h2>La meva biblioteca</h2>
<table border="1">
<tr bgcolor="#9acd32">
<td>Titol</td>
<td>Autor</td>
<td>Editorial</td>
<td>Comentaris</td>
</tr>
<tr>
<td><xsl:value-of select="biblioteca/llibre/titol"/></td>
<td><xsl:value-of select="biblioteca/llibre/autor"/></td>
<td><xsl:value-of select="biblioteca/llibre/editorial"/></td>
<td><xsl:value-of select="biblioteca/llibre/comentaris"/></td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

**Il·lustració 2. Exemple arxiu XSL**



### 2.1.2.2 XSL-FO

*Formatting Objects* (FO) són una implementació específica d'un determinat XSLT, i aplicats a un fitxer XML en proporcionen un altre amb la informació del primer renderitzada a algun format de visualització impresa. Aquesta és la principal diferència entre FO i la resta de XSLT: ambdós transformen un document XML en un altre document XML, però FO està pensat per a generar directament una renderització.

Actualment existeixen 56 FO, la majoria dels quals fan referència a diferents tipus d'àrees rectangulars, i els que no, són contenidors d'àrees rectangulars i espais buits. No tots els FO estan implementats i en alguns casos dependrà molt de l'aplicació que generarà la renderització, ja sigui FOP (aplicació de renderització de FO utilitzada en aquest projecte) o alguna altra.

Per ordre alfabètic els FO són (per anar avançant matèria, entre parèntesis figura si FOP els implementa o no):

*basic-link* (SI), *bidi-override* (NO), *block* (SI), *block-container* (NO), *character* (SI), *color-profile* (NO), *conditional-page-master-reference* (SI), *declarations* (NO), *external-graphic* (SI), *float* (NO), *flow* (SI), *footnote* (NO), *footnote-body* (NO), *initial-property-set* (NO), *inline* (SI), *inline-container* (NO), *instream-foreign-object* (SI), *layout-master-set* (SI), *leader* (SI), *list-block* (SI), *list-item* (SI), *list-item-body* (SI), *list-item-label* (SI), *marker* (NO), *multi-case* (NO), *multi-properties* (NO), *multi-property-set* (NO), *multi-switch* (NO), *multi-toggle* (NO), *page-number* (SI), *page-number-citation* (SI), *page-sequence* (SI), *page-sequence-master* (SI), *region-after* (SI), *region-before* (SI), *region-body* (SI), *region-end* (NO), *region-start* (NO), *repeatable-page-master-alternatives* (SI), *repeatable-page-master-reference* (SI), *retrieve-marker* (NO), *root* (SI), *simple-page-master* (SI), *single-page-master-reference* (SI), *static-content* (SI), *table* (SI), *table-and-caption* (NO), *table-body* (SI), *table-caption* (NO), *table-cell* (SI), *table-column* (SI), *table-footer* (SI), *table-header* (SI), *table-row* (SI), *title* (NO), *wrapper* (NO).

El model utilitzat per FO està basat en caixes rectangulars anomenades àrees, les quals poden contenir text, espai buit o bé altres FO. Cada àrea té un marc i una separació (*padding*) per cada un dels seus costats. La majoria dels FO generen una sola àrea, però a vegades, ja sigui per salts de pàgina, separació de línies, etc... poden generar-ne varies.

Els FO estan dividits formalment en quatre tipus d'àrees rectangulars: contenidors, blocs, línies i "inclusions" (*inline-areas*). Aquestes quatre categories formen una jerarquia molt estricta.

Un contenidor (*container area*) és l'estructura de més alt nivell. Pot ser col·locat en unes coordenades precises dins de l'àrea que el conté. Pot incloure altres contenidors, així com blocs i espais a visualitzar. Per exemple, una pàgina d'un llibre és un contenidor que n'inclou uns altres cinc: la capçalera, el cos, el peu de pàgina i els marges esquerra i dret. Els FO que produeixen contenidors són: *fo:region-body*, *fo:region-before*, *fo:region-after*, *fo:region-start* i *fo:region-end*.

Un bloc (*block area*) representa una estructura a nivell de bloc, com un paràgraf o un element d'una llista. Els blocs són col·locats seqüencialment dins del contenidor que els inclou i poden contenir línies, espais a visualitzar i altres blocs. Els FO que produeixen blocs són: *fo:block*, *fo:list-block*...

Una línia (*line area*) fa referència a cada una de les línies de text dins d'un bloc. Poden contenir incusions així com incusions d'espais (*inline areas* i *inline spaces* respectivament). No existeixen FO que es corresponguin amb les línies, sinó que el motor d'avaluació del formateig les va generant mentre va dividint les línies ja sigui per la mida de la lletra, tipus de justificació escollida....

Finalment, les incusions són part d'una línia, com per exemple cada caràcter, la referència a una nota.... Cada inclusió pot contenir altres incusions o bé incusions d'espais. Els FO que generen incusions són: *fo:inline*, *fo:character*, *fo:page-number* entre altres.

A més de poder definir l'estructura del document generat a partir dels FO, és poden utilitzar propietats per especificar més concretament la renderització que produirà un determinat fitxer. Així per exemple, el format de la pàgina, el color del text o el color de fons de la pàgina, poden ser explicitats mitjançant l'ús dels atributs dels FO. No és casual que moltes de les propietats d'aquests atributs es denominin igual que les utilitzades en les CSS de HTML, de fet, és un repte que no calgui canviar els noms d'aquestes últimes per poder aprofitar l'experiència ja adquirida. No obstant, FO inclou moltes més propietats que CSS, en total més de 200. En aquest cas, igual que amb els FO, no tots els atributs estan implementats. A continuació una llista de les propietats més comunes que estan implementades en el projecte Apache:

*background-color* , *border-after-style* , *border-after-width* , *border-before-color* ,  
*border-before-style* , *border-before-width* , *border-bottom-color* , *border-*  
*bottom-style* , *border-bottom-width* , *border-end-style* , *border-end-width* ,  
*border-left-color* , *border-left-style* , *border-left-width* , *border-right-color* ,  
*border-right-style* , *border-right-width* , *border-start-color* , *border-start-style* ,

*border-start-width , border-top-color , border-top-style , border-top-width , bottom , break-after , break-before , case-title character , color , column-count , column-gap , column-number column-width , country , end-indent , ends-row extent , external-destination , flow-name , font-family , font-size , font-style , font-weight , line-height , margin-bottom , margin-left , margin-right , margin-top , padding-start , padding-top , page-height , page-position , page-width , region-name , text-align , text-align-last , text-decoration , text-indent.*

## **Renderització de documents amb FO.**

Com ja s'ha dit, els FO no són més que una determinada especificació de la forma de visualitzar el contingut d'un fitxer XML escrita en XSL. Per això, normalment, es sol precedir a les definicions de les etiquetes FO amb l'espai de noms *fo*:. Evidentment, aquest espai de noms ha d'estar prèviament declarat. Això s'aconsegueix afegint la sentència següent al fitxer XSL:

```
<xsl:stylesheet
xmlns:fo=http://www.w3.org/XSL/Format/1.0
result-ns="fo">
```

La sentència anterior afegida al principi del document XSL que contindrà la renderització d'un fitxer XML utilitzant FO, permet utilitzar el noms dels FO com ara *fo:block*, *fo:table*...

De la mateixa manera que es pot passar un arxiu XML a HTML, es pot passar a PDF sense res més que posar en el fitxer XSL etiquetes de FO. Posteriorment s'utilitza un programa com FOP per tal d'obtenir la renderització final a PDF. FOP està escrit en Java i pot ser utilitzat tant des de l'interpret de comandes com a través d'algun servidor, com Apache Tomcat.

## **Descripció de pàgines**

L'element arrel d'un FO és *fo:root*. Aquest element conté un *fo:layout-master-set* i zero o més *fo:page-sequence*.

El *fo:layout-master-set* és un contenidor per a les diferents *fo:master-page* que s'utilitzin en el document. Cada *master-page* permet definir el patró per a una pàgina, incloent marges, grandària de la capçalera i peu de pàgina entre altres característiques.

Les *simple-master-pages* es representen amb *fo:simple-master-pages*. La seva funcionalitat és la de definir l'aspecte de cada pàgina patró especificant la mida de les seves regions anterior, posterior, cos, inici i fi. Els FO pertinents serien respectivament: *fo:region-before*, *fo:region-after*, *fo:region-body*, *fo:region-start* i *fo:region-end*. Aquestes regions poden ser emplenades amb contingut utilitzant *fo:flow* o bé *fo:static-content*.

Tres són els atributs principals d'una *simple-master-page*. Aquests són: *page-master-name*, *page-height* i *page-width*. El següent tros de codi mostra la definició d'una *master page* simple de tamany A4 que es pot utilitzar per a incloure el que nosaltres desitgem:

```
<fo:layout-master-set>
  <fo:simple-page-master page-master-name="prova"
    height="29cm" width="21cm" margin-top="0.5cm" margin-bottom="0.5cm"
    margin-left="1.0cm" margin-right="0.5cm">
    <fo:region body/>
  </fo:simple-page-master>
</fo:layout-master-set>
```

Cada document generat amb FO ha de tenir una o més *fo:page-sequences*, cada una de les quals ha de tenir una *fo:sequence-specification* (que indica l'ordre en el que les pàgines patró són utilitzades), zero o més *fo:static-content* amb text a incloure a totes les pàgines, i un *fo:flow* amb dades que s'inclouran a cada pàgina segons es vagin generant.

L'element *fo:flow* ubica el contingut que serà col·locat a les pàgines. Està compost per una seqüència de *fo:block* i altres elements situats al nivell de bloc.

L'element *fo:static* serveix per especificar contingut que sortirà a totes les pàgines. Pot contenir els mateixos elements que *fo:flow*. S'ha de tenir en compte que *fo:static* ha d'aparèixer abans que l'element *fo:flow* al definir la seqüència d'elements de la pàgina. En aquest petit exemple que segueix a continuació es pot veure un *fo:static* que inclou un advertiment de *copyright* i s'encarrega de numerar totes les pàgines.

```
<fo:static-content flow-name="xsl-before">
  <fo:block>(C) PEX 2007</fo:block>
  <fo:block text-align-last="centered" font-size="12pt">
    <fo:page-number/>
  </fo:block>
</fo:static-content>
```

Els *fo:block* poden estar continguts en *fo:flow*, *fo:static-content* o bé en un altre *fo:block*. Un *fo:block* pot contenir, per exemple, text pla, un altre *fo:block*, *fo:inline*...

```
<fo:block>
  <fo:inline font-style="italic">
    Això està amb cursiva
  </fo:inline>
  Pàgina: <fo:page-number/>
  <fo:inline>
    Això no ho esta!!
  </fo:inline>
</fo:block>
```

### 2.1.3 APIs de Java per a treballar amb XML.

Un cop tenim el document XML amb la informació o bé en volem crear un amb informació generada per nosaltres, com s'utilitza o es trasllada aquesta informació a un programa Java?. Els encarregats de realitzar aquesta tasca són els analitzadors (*parsers*). Els analitzadors més estesos en Java per a treballar amb XML són: SAX, DOM i JDOM.

#### 2.1.3.1 SAX

SAX (Simple Api for XML) processa el document o informació en XML d'una manera molt diferent a DOM, SAX processa la informació per events. A diferència de DOM, que genera un arbre jeràrquic en memòria, SAX processa la informació en XML conforme aquesta sigui presentada (event per event).

Les seves característiques principals són:

- SAX és un analitzador (*parser*) ideal per a manipular arxius de mida gran, ja que no requereix de molta memòria (no genera cap arbre com en el cas de DOM).
- És més ràpid i més senzill d'utilitzar que DOM.
- Per contra, la simplicitat té un preu, degut a que SAX funciona per events, no es possible manipular informació un cop processada, en DOM no existeix aquesta limitació doncs al generar un arbre jeràrquic en memòria ens podem moure pels seus nodes i d'aquesta forma accedir a la informació quan es vulgui.

L'especificació SAX més recent és la 2.0 i la majoria de *parsers* SAX disponibles ja la implementen.

### **2.1.3.2 DOM**

DOM ( *Document Object Model*) genera un arbre jeràrquic en memòria. A cada element del document XML, és considerat un node dins de l'arbre. Aquest arbre jeràrquic en memòria permet que a través del *parser* es pugui manipular la informació.

Les seves principals característiques són:

- Es pot afegir un node en qualsevol punt de l'arbre.
- Es pot eliminar la informació d'un node en qualsevol punt de l'arbre.
- Permet crear documents XML.

Degut a que DOM és només una especificació, existeixen diferents analitzadors DOM. L'especificació DOM més recent és la 3.0 i la majoria d'analitzadors DOM disponibles ja la implementen.

### **2.1.4 Eines de software per a treballar amb XML.**

En realitat, per a treballar amb XML, no fa falta cap editor especial, però va bé un editor específic de XML o inclús un entorn integrat que ajudi a identificar bé el codi, tancar les etiquetes en l'ordre correcte o que tregui l'etiqueta i atributs admissibles en cada moment en funció del DTD. Si es treballa en Windows una bona opció és el eXcelon Stylus o UltraEdit; en Linux es pot treballar bé amb el XEmacs que valida utilitzant un DTD si és necessari.

De tota manera, són moltes les eines que van apareixent per a treballar amb XML. Una mostra més de que es tracta d'una tecnologia relativament nova i que cada cop compta amb el suport de més companyies del món de la informàtica. Per exemple, Microsoft, en la seva última versió de la seva suite d'eines ofimàtiques Office, ja inclou suport XML en el seu editor de texts Word. L'usuari és lliure d'escollir l'opció en la que se senti més còmode i millor li convingui a l'hora de treballar amb XML.

## 2.2 Java Web Start

Java Web Start és la implementació de referència de la especificació JNLP (Java Networking Launching Protocol) i està desenvolupada per Sun Microsystems, mitjançant la qual es possible executar aplicacions Java que estiguin en un servidor web d'aplicacions comprovant prèviament si el client té la versió actualitzada de l'aplicació. Si no es així, es descarregarà la darrera versió i s'executarà en local. L'execució d'aquest tipus d'aplicacions pot ser efectuat mitjançant enllaços en una pàgina web o bé a través d'un enllaç a l'escriptori del client.

Així doncs els punts forts de Java Web Start són:

- Permet activar aplicacions remotes amb un sol clic.
- Garanteix que s'està executant la darrera versió de l'aplicació.
- Elimina complexos procediments d'instal·lació i d'actualització.

Actualment Java Web Start es distribueix juntament amb el JRE (Java Runtime Environment).

Els fitxers que contenen la informació sobre on es troba l'aplicació, la versió,... tenen l'extensió `jnlp`.

Un arxiu `jnlp` és el nucli de la tecnologia JNLP i és tracta d'un arxiu XML amb una determinada sintaxi que es pot observar en el següent exemple:

```
00 <?xml version="1.0" encoding="utf-8"?>
01 <!-- JNLP File for Demo Application -->
02 <jnlp
03   spec="1.0+"
04   codebase="http://java.sun.com/apps"
05   href="example.jnlp">
06   <information>
07     <title>Demo Application</title>
08     <vendor>Sun Microsystems.</vendor>
09     <homepage href="docs/help.html"/>
10     <description>This is a Demo Application</description>
11     <description kind="short">A demo.</description>
12     <icon href="images/demologo.gif"/>
13     <offline-allowed/>
14   </information>
15   <security> <all-permissions/>
16 </jnlp>
```

```
17 <resources>
18   <j2se version="1.3+"/>
19   <jar href="example.jar"/>
20 </resources>
21 <application-desc main-class="Example"/>
22 </jnlp>
```

### Il·lustració 3. Exemple arxiu jnlp

Com es pot veure, l'element principal és *jnlp* (línia 02). Els atributs *codebase* i *href* d'aquest element són importants, doncs *codebase* indica la URL base que es prendrà com a referència per a totes les direccions especificades en els atributs *href* dels diferents elements i l'atribut *href* d'aquest element indica la localització de l'arxiu jnlp.

L'element *information* (línia 06) defineix informació general sobre l'aplicació com el seu títol, autor, descripció...

L'element *security* (línia 15) defineix el nivell de seguretat que es desitja quan l'aplicació s'executi a la màquina client. Hi ha 2 nivells de seguretat: *j2ee-application-client-permissions* i *all-permissions*. Si s'especifica el nivell de seguretat *all-permissions* tots els arxius JAR hauran d'anar signats i el primer cop que un client executi l'aplicació es mostrarà un certificat que aquest haurà d'acceptar.

L'element *resources* (línia 17) indica quins arxius constitueixen l'aplicació.

L'element *application-desc* (línia 21) bàsicament especifica quina és la classe principal (la que conté el mètode Main).

## 2.3 SSL

Secure Socket Layer (SSL) és una tecnologia, concretament un protocol que utilitza tècniques criptogràfiques i proporciona comunicacions segures a través de la xarxa. Aquest protocol va ser dissenyat i desenvolupat per Netscape i proporciona autenticació i privacitat tot permetent a les aplicacions client-servidor comunicar-se d'una forma dissenyada per tal de prevenir les possibles escoltes del missatge, la falsificació de la



identitat del remitent i mantenir la integritat del missatge. SSL s'executa en una capa entre els protocols d'aplicació com HTTP i sobre el protocol de transport TCP.

SSL implica una sèrie de fases bàsiques:

- Negociar entre les dues parts l'algorisme que s'utilitzarà en la comunicació.
- Intercanvi de claus públiques i autenticació basada en certificats digitals.
- Xifrat del tràfic basat en un xifrat simètric.

En la primera fase de negociació de l'algorisme criptogràfic a utilitzar en la comunicació entre el client i el servidor, l'especificació actual proporciona les següents opcions:

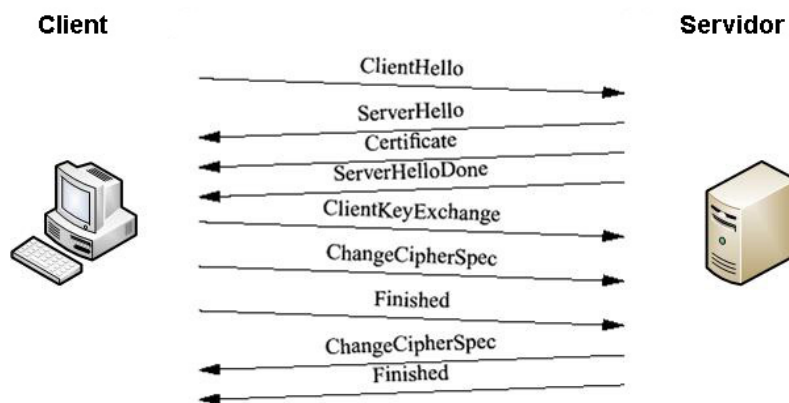
- Per criptografia de clau pública: RSA, Diffie-Hellman, DSA (Digital Signature Algorithm) o Fortezza.
- Per criptografia de clau privada: RC2, RC4, IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard), Triple DES o AES (Advanced Encryption Standard).
- Amb funcions hash: MD5 o de la família SHA.

El protocol SSL intercanvia registres. Opcionalment, aquests registres poden ser comprimits, xifrats i empaquetats amb un codi d'autenticació de missatge (MAC).

Quan s'inicia la connexió s'inicia el protocol de handshake entre client i servidor. El client envia i rep diverses estructures handshake:

1. El client envia al servidor un missatge ClientHello especificant una llista de xifratges, mètodes de compressió i la versió del protocol SSL més alta permesa. També envia bytes aleatoris (anomenats Challenge) que s'utilitzaran més tard i a més pot incloure l'identificador de la sessió.

2. El client rep un registre ServerHello, en el que el servidor escull els paràmetres de la connexió a partir de les opcions oferides en anterioritat pel client.
3. Amb el paràmetres de la connexió resultats, client i servidor intercanvien certificats.
4. Client i servidor negocien una clau secreta comuna anomenada master secret, possiblement utilitzant el resultat d'un intercanvi Diffie-Hellman, o simplement xifrant una clau secreta amb una clau pública que és desxifra amb la clau privada de cada un. Totes les restants dades de clau són derivades a partir d'aquest master secret i els valors aleatoris (Challenge) anteriorment generats en el client.



Il·lustració 4. SSL handshake

### 2.3.1 Creació d'un certificat digital SSL

El certificat digital és un sistema format per un conjunt de requisits: clau pública, clau privada i entitat de certificació. Fa possible garantir la identitat de l'autor d'un missatge o transacció electrònica.

Aquest certificat ha de ser emès per una Autoritat de Certificació, entitat de confiança encarregada de verificar les identitats certificades.

L'ús del certificat digital garanteix:

- La identificació de l'emissor. Només pot haver-lo enviat la persona que el signa.
- La integritat de la transacció. S'ha rebut tota la informació i no ha estat manipulada per cap persona no autoritzada.
- La irrefutabilitat dels compromisos adquirits per via electrònica. La signatura electrònica té el mateix valor legal que la firma gràfica, amb la qual cosa els compromisos no poden ser rebutjats.
- La confidencialitat de l'enviament. Una comunicació xifrada només pot ser llegida pel seu destinatari final i no pot haver estat manipulada.

El certificat digital permet al seu titular:

- Identificar-se eficaçment en línia.
- Signar documents amb validesa legal mitjançant Internet.
- Assegurar que la seva identitat no és suplantada.
- Protegir la informació tramesa.
- Garantir la integritat de la comunicació entre ambdues parts.

Hi ha moltes maneres de crear un certificat digital SSL. La que aquí s'explica és la que utilitza l'eina *keytool*, que proporciona Java, per crear aquest tipus de certificats.

L'eina *keytool* es troba juntament amb la resta d'executables de la plataforma Java, concretament a la carpeta *bin* del JDK que es tingui instal·lat i funciona mitjançant línia de comandes.

La seva sintaxis es pot observar en el següent exemple:

```
keytool -genkey -keystore arxiu.cer -keyalg RSA
```

Després de preguntar-nos la clau que s'utilitzarà per generar el Keystore i les dades del certificat (nom, organització, població...), aquesta ordre crearia el certificat *arxiu.cer* utilitzant les dades subministrades i amb claus RSA.

## Capítol 3

# Desenvolupament de la plataforma

Els requeriments inicials establerts, la temporització de les diferents etapes de desenvolupament, l'arquitectura i el disseny de la plataforma són els punts que es tracten en aquest capítol.

Com es pot veure, les decisions de desenvolupament han anat sempre guiades pels requeriments inicials i també per la intenció de que la plataforma fos el més simple d'utilitzar i d'instal·lar sense renunciar a les més elevades opcions de seguretat.

El fet de tenir una instal·lació i funcionament fàcil es important degut a certs tipus de programes que sovint es troben a les aules d'informàtica i que dificulten la instal·lació de programari nou a les màquines.

Pel mateix motiu, també s'ha procurat que la plataforma pogués funcionar sense haver de tenir privilegis d'administrador.

### **3.1 Requeriments inicials**

Bàsicament, els requeriments inicials que es van marcar van ser:

- Desenvolupar una plataforma per simplificar tot el procés de realització d'exàmens (crear, desplegar en xarxa i corregir un examen).
- Aplicació multi plataforma.
- Diferents tipus de preguntes.
- Garantir l'autenticació i la seguretat de les dades.
- Facilitat d'instal·lació i ús.

Un cop els requeriments inicials anaven prenent cos, s'anaven marcant nous objectius:

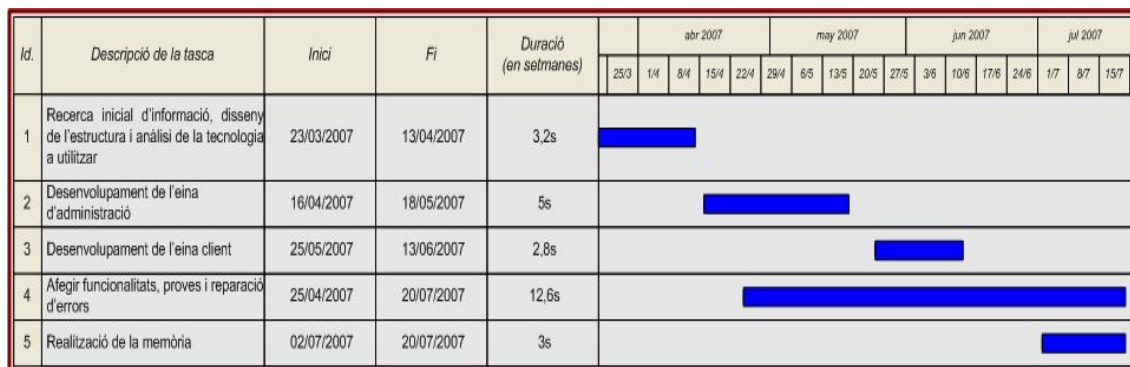
- Versatilitat de les dades.
- Transformacions a HTML i PDF.
- Generació de diferents tipus d'informes de resultats.

La intenció final de la plataforma és la de simplificar tot el procés de realitzar un examen, garantint l'autenticació, seguretat i integritat de les dades juntament amb un ús fàcil i simple per a la majoria de possibles usuaris i situacions que es puguin derivar.

També s'ha procurat que la plataforma fos molt visual amb el disseny d'una interfície gràfica que mostra en cada moment l'estat de les diferents accions realitzades.

### 3.1 Temporitziació

La temporització prevista per al desenvolupament de la plataforma i que, més o menys, s'ha anat complint, ha estat la que es veu representada en el següent diagrama de Gantt:



Il·lustració 5. Temporitziació del projecte

La dedicació diària al desenvolupament del projecte no era total, però es podria fixar en una mitja de 15 hores a la setmana. El projecte ha estat realitzat en una mica més de temps que el que s'estableix en nombre de crèdits corresponents, bàsicament degut a la complexitat d'algunes operacions i les moltes proves que s'han tingut que realitzar pel bon funcionament de la plataforma.

### 3.2 Estudi de mercat

Una de les primeres coses que es va fer abans de començar a desenvolupar la plataforma va ser realitzar un estudi de mercat per tal de comprovar si ja existien aplicacions similars a la que es volia fer. Al buscar opcions similars per Internet, no s'he n'han trobat moltes.

Aquestes són algunes de les aplicacions amb funcions similars als objectius d'aquest projecte:

#### TCEXAM

TCEXAM és una eina per a generar i manipular tests i exàmens on-line. Bàsicament és tracta d'un script PHP que, amb l'ajuda d'una base de dades, controla i realitza les proves.

Un dels punts dolents d'aquest programa és que la instal·lació i posada en funcionament no resulta gens fàcil. S'ha de tenir un cert grau de coneixements informàtics per poder realitzar amb èxit els primers passos, ja que s'ha de configurar PHP adequadament.

TCEXAM és independent de la plataforma on s'executa, doncs principalment està format per scripts de PHP i utilitza MySQL o PostgreSQL, dos SGBD multi plataforma. Sense cap mena de dubte aquesta és la millor aplicació que s'ha trobat i la que més s'ajusta al que es vol fer.

#### **Academic Test Tool 3.0**

Aquesta és una altra eina similar, però només és per sistema operatiu Windows, per tant, no és multi plataforma. Tampoc és gratuït, és poc flexible i no es poden realitzar exàmens en xarxa.

#### **Self Test i TestGIP**

No són multi plataforma. Només permet utilitzar qüestions del tipus veritat o fals.

SelfTest no pot realitzar exàmens en xarxa.

#### **Moodle**

Moodle és un paquet de programari lliure per a la creació de cursos i llocs web basats en Internet. És un projecte en desenvolupament dissenyat per a donar suport a un marc d'educació constructivista social.

Moodle es distribueix de manera gratuïta com a programari lliure (Open Source) (sota la Llicència Pública GNU). Bàsicament això vol dir que Moodle té drets d'autor (copyright), però que teniu algunes llibertats. Podeu copiar, utilitzar i modificar Moodle sempre que accepteu: proporcionar el codi font als altres, no modificar o eliminar la llicència original i els drets d'autor, i aplicar aquesta mateixa llicència a qualsevol treball que en derivi.

Moodle és multi plataforma, permet fer exàmens en xarxa, disponible en diferents idiomes i permet realitzar diferents tipus de qüestions amb l'ajuda de plugins addicionals creats per diferents programadors, però més que a fer exàmens està orientat a la realització de cursos on-line.

El resum de les característiques d'aquestes aplicacions es pot observar en la següent taula:

NOM	MULTI PLATAFORMA	EXAMENS EN XARXA	DIFERENTS PREGUNTES	FÀCIL INSTAL·LACIÓ	SEGURETAT SSL
TCEXam	SI	SI	SI	NO	NO
Academic Test Tool	NO	NO	NO	SI	NO
Self Test	NO	NO	SI	SI	NO
TestGIP	NO	SI	NO	NO	NO
Moodle	SI	SI	SI	NO	NO
PEX	SI	SI	SI	SI	SI

**Il·lustració 6. Taula de característiques**

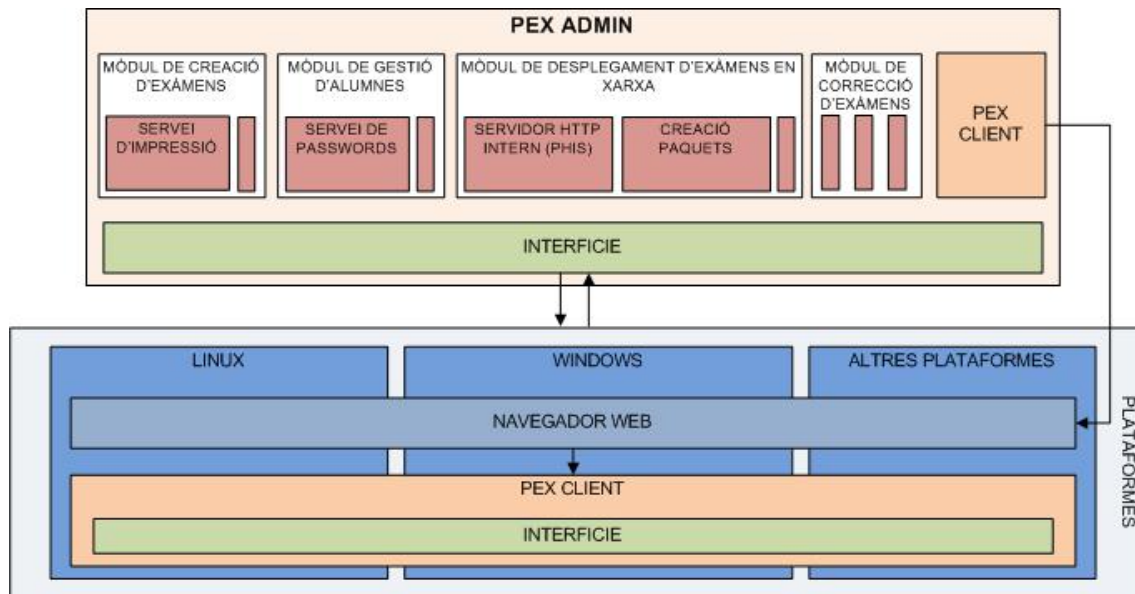
### **3.3 Arquitectura de la plataforma**

El llenguatge de programació utilitzat principalment és Java degut a les seves característiques de programari lliure, codi obert, orientat a objectes, permet crear aplicacions distribuïdes i és multi plataforma. Aquestes característiques compleixen perfectament amb els requeriments inicials establerts. S'ha utilitzat NetBeans IDE 5.5 com a plataforma pel desenvolupament de les aplicacions.

La idea principal és la d'establir una arquitectura client-servidor a l'hora de realitzar els exàmens en xarxa. De tal forma que l'aplicació d'administració (PEX Admin) sigui la que controla tot el procediment, des de la creació i desplegament (distribuir, controlar i recollir els resultats) d'un examen fins a l'obtenció dels resultats, tot vetllant per tal de garantir la seguretat i la integritat de les dades, i l'aplicació client (PEX Client) sigui l'encarregada de realitzar l'examen pròpiament dit.



L'arquitectura de la plataforma vindria representada en el següent gràfic:



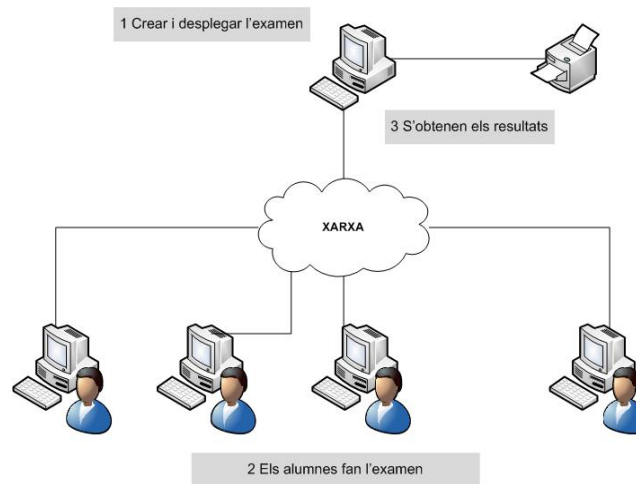
**Il·lustració 7. Arquitectura de l'aplicació**

La interfície que interacciona amb els mòduls utilitza el patró model-vista. Molts dels elements que presenta la llibreria Swing de Java i que utilitza la interfície, com són les taules, llistes desplegable... ja l'utilitzen. Aquest patró d'arquitectura de software separa les dades d'una aplicació de la interfície de l'usuari.

La part de la creació d'exàmens es farà mitjançant la interfície en Java que permetrà definir tant les preguntes com les opcions de l'examen. Els exàmens i altres dades utilitzades en el cicle de realització d'un examen es guarden en format XML i es poden recuperar en qualsevol moment.

Quan s'ha acabat de crear l'examen, es pot escollir entre distribuir l'examen per a la seva posterior realització en xarxa o bé guardar-lo, ja sigui en format de la pròpia plataforma o bé exportant-lo a un document PDF per a la seva posterior utilització.

Un cop finalitzat l'examen, es pot procedir a l'anàlisi dels resultats obtinguts. Aquesta part també la realitzarà la mateixa aplicació que s'ha fet servir per crear l'examen.

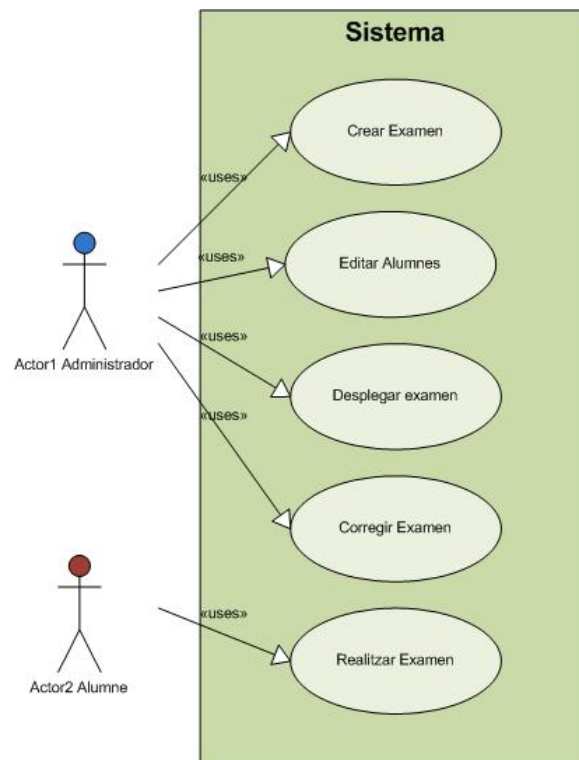


Il·lustració 8. Cicle vital d'un examen

### 3.4 Disseny de la plataforma

El disseny de la plataforma gira entorn als requeriments que es van especificar i que s'han procurat complir.

A continuació es detalla el diagrama de casos d'ús de l'aplicació representant els possibles actors i la funcionalitat que ofereix el sistema en el que a interacció externa es refereix.



Il·lustració 9. Diagrama de casos d'ús

S'ha optat per un disseny modular de cada un dels que podrien ser els grans punts forts de la plataforma. Acte seguit es comentaran amb més detall cada un d'aquests mòduls.

### 3.4.1 Mòdul de creació d'exàmens

Per a la manipulació de les dades que utilitza la plataforma s'utilitza XML. Un examen no es res més que un document XML representat en memòria per un arbre DOM. Un document d'examen presenta l'estructura que es pot veure en el següent exemple:

```
<?xml version="1.0"?>
<examen>
  <pregunta numero="1" tipus="1" puntuacio="1">
    <enunciat>Quina és l'arrel quadrada de 49?</enunciat>
    <opcio>3</opcio>
    <opcio>5</opcio>
    <opcio>7</opcio>
    <opcio>9</opcio>
  </pregunta>
  <pregunta numero="2" tipus="2" puntuacio="1.5" linies="6" imatge="img2.jpg">
    <enunciat>Descriu el conjunt dels nombres reals</enunciat>
  </pregunta>
  <pregunta numero="3" tipus="3" puntuacio="1">
    <opcio>el reste de dividir 8 entre 3 és 2</opcio>
    <opcio>multiplicar 2 per 5 és 12</opcio>
    <opcio>dividir 16 entre 4 és 4</opcio>
    <opcio>sumar 3 a 5 és 9</opcio>
    <opcio>restar 10 a 20 dona 10</opcio>
  </pregunta>
  ...
  ...
</examen>
```

Il·lustració 10. Estructura d'un document d'examen

Com es pot veure, l'element principal és *<examen>* i d'aquest element tenim tants elements *<pregunta>* com preguntes tingui l'examen. L'element pregunta presenta un conjunt d'atributs identificadors: el número de pregunta, el tipus de pregunta del que es tracta i la puntuació són atributs comuns a tots els tipus de pregunta. En funció del tipus de pregunta o de si s'utilitza alguna imatge auxiliar hi haurà, a més a més, presents aquests dos atributs.

Cada element *<pregunta>* pot tenir un element *<enunciat>* i/o un o més d'un elements *<opcio>* que serveixen per especificar l'enunciat de la pregunta i les possibles opcions de resposta a la pregunta. La presència d'aquests elements dependrà del tipus de pregunta utilitzat. El tipus de pregunta 1 significa que es tracta d'una pregunta de tipus

test, per tant, l'element <pregunta> tindrà un element <enunciat> i tants elements <opcio> com hagi especificat l'usuari com a possibles respostes a la pregunta, el tipus 2 significa que es tracta d'una pregunta de descripció i el tipus 3 fa referència a una pregunta de veritat o fals.

Cada cop que s'afegeix és modifica o s'elimina una pregunta, es visualitza una transformació a HTML d'aquest arbre DOM utilitzant el full d'estil XSL seleccionat a les propietats de l'examen.

Aquest arbre DOM es pot transformar a un document PDF, guardar en un arxiu i recuperar-lo quan així es desitgi per tal de modificar-lo o desplegar-lo en xarxa.

També es pot realitzar la transformació a PDF del document. Per més informació sobre el tema es pot consultar el següent capítol en el que es presenta el manual d'usuari de la plataforma.

### 3.4.2 Mòdul d'edició d'alumnes

Les dades dels alumnes també es manipulen amb XML. Igual que els exàmens, per manipular els alumnes s'utilitza una estructura d'arbre DOM. La seva estructura es pot veure a continuació:

```
<?xml version="1.0"?>
<alumnes>
  <propietats assignatura="PI" curs="2006/07" />
  <alumne Clau="YnmIb94JCzA48=" Cognom1="Soler" Cognom2="Ribes" DNI="43745674"
Nom="Joan" />
  <alumne Clau="YnmIb94JCzA48=" Cognom1="Garcia" Cognom2="Farré" DNI="78042324"
Nom="Maria" />
  <alumne Clau="YnmIb94JCzA48=" Cognom1="Vila" Cognom2="Gracia" DNI="78034564"
Nom="Gerard" />
  <alumne Clau="YnmIb94JCzA48=" Cognom1="Fàbrega" Cognom2="Brugulat" DNI="43734343"
Nom="Gemma" />
  <alumne Clau="YnmIb94JCzA48=" Cognom1="Mas" Cognom2="Morelló" DNI="78023434"
Nom="Jordi" />
  <alumne Clau="YnmIb94JCzA48=" Cognom1="Novau" Cognom2="Cases" DNI="43687878"
Nom="Pere" />
  ...
  ...
</alumnes>
```

Il·lustració 11. Estructura d'un document d'alumnes

L'element principal de l'arbre és *<alumnes>*, i aquest conté tants elements *<alumne>* com alumnes hi hagi. Cada element *<alumne>* està format per 5 atributs: el primer cognom, el segon cognom, el nom, el DNI i la clau d'accés que haurà d'utilitzar l'alumne a l'hora de realitzar un examen.

Per tal de maximitzar la seguretat, la clau d'accés es guarda a l'arbre ja xifrada utilitzant una funció hash d'una sola direcció amb un algorisme SHA-1 de 160 bits. Aquesta transformació la realitza el mòdul de servei de passwords de la plataforma que utilitza les llibreries de Java que proporcionen aquesta funcionalitat.

### 3.4.3 Mòdul de desplegament en xarxa

Juntament amb el de corregir, és un dels mòduls més complexes. En aquest mòdul es vincula un document d'examen amb un d'alumnes i s'estableixen propietats referents a l'examen i al desplegament pròpiament dit. Aquesta vinculació de documents i altres atributs queda palesa en un nou arbre DOM que serà guardat com un fitxer (*deploy.xml*) i que contindrà totes les peculiaritats de l'examen que s'acaba de desplegar.

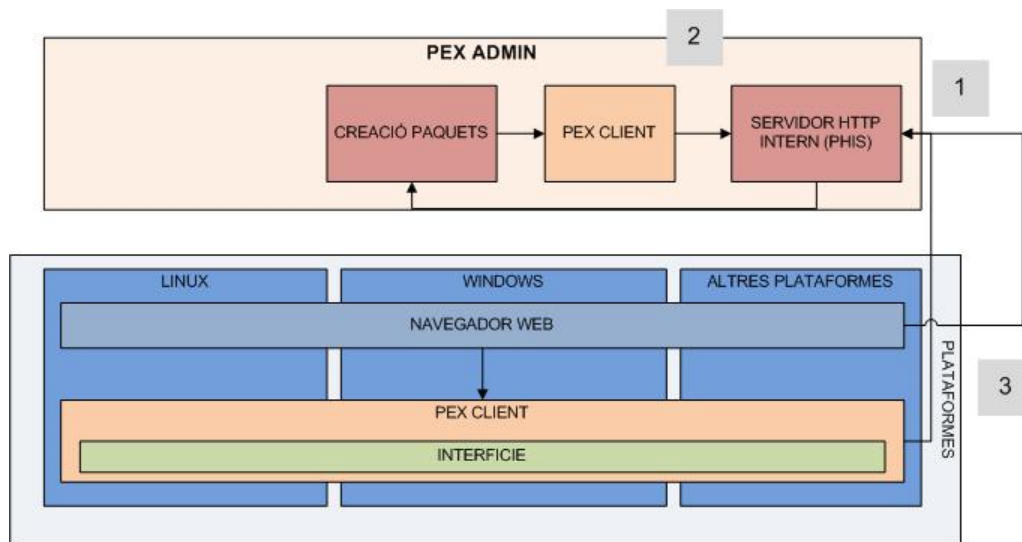
El fet de desplegar un examen en xarxa implica tenir una forma de distribuir l'examen als clients. Si es desitja que la plataforma requereixi el mínim d'instal·lació possible i facilitat d'ús, com així és, el mètode finalment escollit ha de facilitar, com a mínim, aquestes opcions.

Es per això que en aquest mòdul s'ha optat, per una banda, implementar un servidor HTTP intern a l'eina d'administració per tal de poder donar aquest servei i per una altra utilitzar la tecnologia Java Web Start per tal de distribuir l'aplicació client encarregada de la realització dels exàmens. Aquest mètode no requereix cap tipus d'instal·lació a l'ordinador client.

Els passos que es segueixen per tal de descarregar l'aplicació client a un client són els següents:

1. El servidor HTTP intern rep una petició del client des d'un navegador web. Comprova la seva identitat i en cas d'una identificació correcta continua.

2. Utilitzant el servei de creació de paquets de la plataforma, s'empaqueta l'aplicació client juntament amb l'examen desplegat i l'arxiu d'informació *deploy.xml*.
3. Mitjançant Java Web Start es desplega l'aplicació al client.

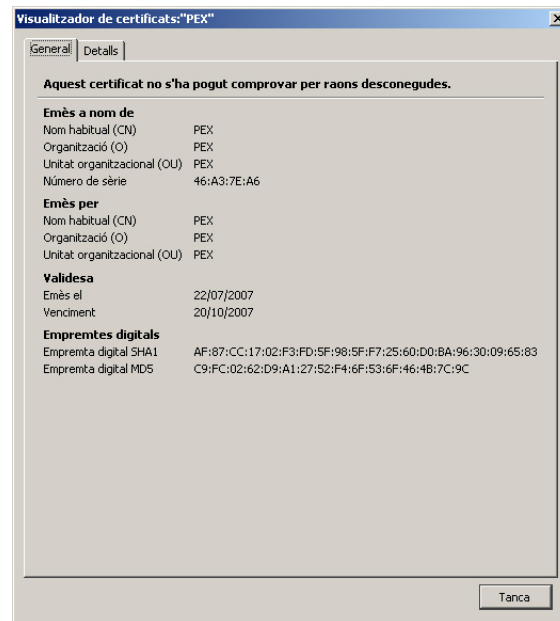


Il·lustració 12. Procés de desplegament en xarxa

El servidor HTTP pot atendre peticions de forma simultània ja que cada petició s'atén en un fil de procés en paral·lel. Cada nova petició crea un *Thread* per atendre-la.

A partir del moment inicial en el que el servidor HTTP rep la petició d'un client a través del navegador web ja es tracta d'una connexió segura mitjançant el protocol SSL 3.0. Tot el procés de comunicació entre l'administrador i el client també es fa sota aquesta connexió segura.

Cada aplicació disposa del seu pertinent certificat digital, creat mitjançant l'eina *keytool* tal i com s'indica en el capítol anterior, sense el qual no es possible comunicar-se amb l'aplicació d'administració.



**Il·lustració 13. Certificat SSL**

L'arxiu jnlp que envia el servidor web intern per desplegar l'aplicació client té la següent estructura:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!-- JNLP File for PEX Client -->
<jnlp spec="1.0+" codebase="phisp" href="client.jnlp">
<information>
  <title>PEX [Client]</title>
  <vendor>PEX</vendor>
  <homepage href="http://www.google.com"/>
  <description>PEX [Client]</description>
  <description kind="short">Programa client per a PEX</description>
  <offline-allowed/>
</information>
<resources>
  <j2se version="1.6+"/>
  <jar href="PEX__Client_.jar"/>
  <argument>paramdni</argument>
  <argument>paramcognom1</argument>
  <argument>paramcognom2</argument>
  <argument>paramnom</argument>
</resources>
<application-desc main-class="pexclient.Main"/>
</jnlp>
```

**Il·lustració 14. Arxiu jnlp de desplegament**

De la recollida dels resultats també se n'encarrega aquest mòdul. Novament, els resultats es representen en XML amb el seu pertinent arbre DOM en memòria per a la seva manipulació. Posteriorment, els resultats són guardats en un arxiu per al seu posterior tractament en el següent mòdul de correcció. L'estructura de cada resposta es molt semblant a la de l'examen substituint el contingut de cada opció per la resposta que ha posat l'usuari.

#### 3.4.4 Mòdul de correcció d'exàmens

Aquest darrer mòdul de correcció d'exàmens genera a partir d'una plantilla correctora que conté les respostes, que igual que en els anteriors casos està en format XML i representada en memòria per un arbre DOM, un document XML de correccions amb les qualificacions detallades de cada alumne. A partir d'aquest darrer document es generen les pertinents transformacions a HTML, PDF i CSV.



Il·lustració 15. Procés de correcció

L'esquema d'un document corrector és el mateix que el d'una resposta d'un alumne, però en aquest cas, com es obvi, conté les respostes correctes a les preguntes de l'examen.

Una pregunta de tipus test d'un resultat es considera correcta si i només si, totes les opcions tenen el mateix valor de veritat que les de la pregunta del document corrector. D'aquesta manera es corregeix correctament una pregunta de tipus test de selecció simple i múltiple.

Una pregunta de descripció es correcta si i només si, coincideix exactament amb el contingut que s'ha introduït al document corrector.

Una pregunta de tipus veritat o fals es correcta si i només si, els dos valors de veritat coincideixen exactament amb els que figuren a la pregunta del document corrector.

Les preguntes de tipus test i veritat o fals poden estar subjectes a penalitzacions per error. Aquestes penalitzacions es poden establir de dues formes diferents:

- Ràtio manual. S'aplica una penalització en base a la ràtio establerta per l'administrador (consultar el manual d'usuari).



- Ràtio calculada. Mitjançant aquesta opció es calcula automàticament la penalització que s'ha d'aplicar basada en que l'esperança de la nota d'un alumne que contesti a l'atzar sigui 0. Per exemple, en el cas d'una pregunta de tipus test que val 2 punts i amb 5 possibles opcions, una de les quals és la correcta, aquesta penalització vindria expressada per la següent fórmula:

$$0 = 2 * \left(\frac{1}{5}\right) - k * \left(\frac{4}{5}\right) \text{ on } k \text{ es la penalització a aplicar}$$

En el document de correccions es guarda detalladament cada una de les correccions realitzades, la nota obtinguda en cada pregunta, els possibles descomptes deguts a penalització per error en determinades preguntes... L'esquema d'aquest document de correccions és pot veure en la següent imatge o bé, més detalladament, en el cd-rom annexat, editant qualsevol dels arxius de resultats dels exàmens de mostra.

```
<?xml version="1.0" encoding="UTF-8" ?>
<correccions>
  <propietats linia1="IES Joan Brudieu" linia2="Instal·lació i manteniment de sistemes informàtics"
linia3="22/09/2007 (1 Parcial, 1 Avaluació)" />
  <correccio Cognom1="Novau" Cognom2="Cases" Nom="Pere" DNI="43687878" >
    <correc numero="1" correcta="SI" corregida="SI" nota="0.5" punts="0.5" tipus="1" />
    <correc numero="2" correcta="SI" corregida="SI" nota="0.5" punts="0.5" tipus="1" />
    ...
  </correccio>
  <correccio Cognom1="Fàbrega" Cognom2="Cases" Nom="Gerard" DNI="78047878" >
    <correc numero="1" correcta="NO" corregida="SI" nota="0" punts="0.5" tipus="1" />
    ...
  </correccio>
</correccions>
```

**Il·lustració 16. Estructura d'un document de correccions**

### 3.4.5 Altres consideracions de disseny

Per a la realització de la interfície s'han utilitzat les llibreries Swing de Java.

També considerar que tots els programes utilitzats en el desenvolupament de la plataforma són programari lliure i les diferents tecnologies emprades són multi plataforma, d'aquesta manera no es limita l'aplicació en cap sentit.

Una transformació que produeixi un document HTML, XML, CSV... es realitza mitjançant l'ús de patrons i per una transformació, per exemple, a PDF que ja implica haver de fer una renderització i el conseqüent ús que això implica d'un motor de renderització que suporti XSL-FO, que en l'aplicació que ens ocupa, aquest motor és FOP.

Juntament amb la plataforma, es proporcionen dos tipus de transformacions per a la visualització dels exàmens: la vista 'clàssica' i la 'moderna'. Amb la informació proporcionada sobre l'estructura dels documents XML en aquest capítol, resulta fàcil modificar o crear noves transformacions que s'adaptin a qualsevol cas.

## Capítol 4

# Manual d'usuari

A continuació es presenta el manual d'usuari de la plataforma per tal de facilitar la tasca de crear, desplegar i corregir exàmens a tots aquells usuaris que hi estiguin interessats.

Es tracta d'un arxiu PDF que descriu tots els processos que es poden realitzar amb les eines que proporciona PEX. Els requeriments mínims, la instal·lació de l'aplicació, com crear un examen, com desplegar-lo en xarxa, com efectuar degudament la correcció d'un examen són alguns dels punts que es tracten en aquest manual.

Aquí en aquesta documentació s'adjunta la primera versió del manual però es possible que l'aplicació contingui una revisió més recent ja que està en constant actualització.

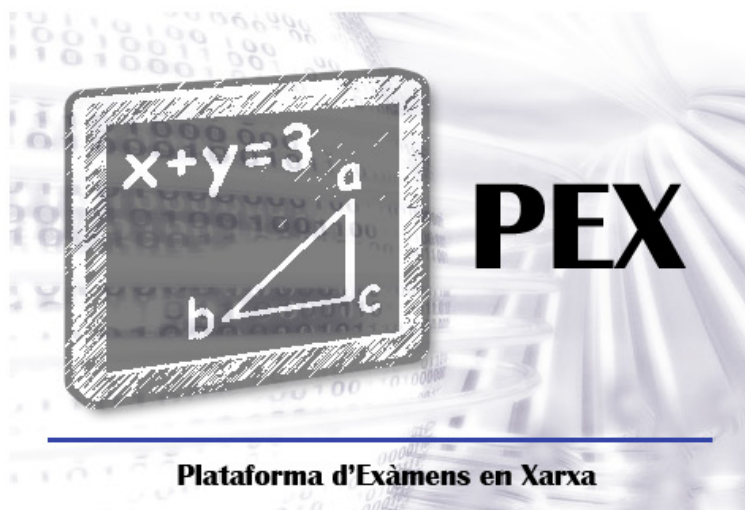
El present manual també està inclòs a la plataforma en forma d'ajuda a l'usuari.

# PEX 1.0

## Manual d'usuari

---

Revisió: 1.0  
14/07/2007



## Índex

1 Introducció.....	3
2 Entrant a PEX .....	4
2.1 Requeriments .....	4
2.2 Instal·lació .....	4
2.3 Accés a PEX .....	6
3 Funcionament de PEX Admin.....	7
3.1 Elements de la pantalla inicial.....	7
3.2 Crear o modificar un arxiu d'examen.....	9
3.2.1 Propietats de l'examen.....	10
3.2.2 Afegir pregunta.....	12
3.2.3 Modificar una pregunta .....	14
3.2.4 Eliminar una pregunta .....	14
3.3 Crear o modificar un arxiu d'alumnes .....	15
3.4 Desplegar un examen en xarxa.....	17
3.4.1 PEX Client.....	19
3.5 Corregir un examen .....	21
3.6 Darreres consideracions.....	24
5 Annexes .....	25
5.1 Captures de l'aplicació en funcionament.....	25
5.2 Llistat dels errors més freqüents.....	27
5.3 Taula d'il·lustracions .....	28

## 1 Introducció

Aquest manual explica com utilitzar la plataforma creada per a la gestió d'exàmens en xarxa PEX i està adreçat a usuaris amb coneixements bàsics d'informàtica.

L'objectiu principal de PEX és el de simplificar notablement la tasca de crear, realitzar i corregir exàmens a totes aquelles persones que per la seva tasca laboral hagin de realitzar aquest tipus de proves.

La plataforma consta de dos aplicacions, l'aplicació d'administració i l'aplicació client. L'aplicació d'administració crea, desplega i corregeix exàmens així com gestiona els alumnes que duren a terme el mateix. L'aplicació client s'utilitza per a la realització dels exàmens per part dels alumnes.

## 2 Entrant a PEX

### 2.1 Requeriments

Com es tracta d'una eina multi plataforma, els requeriments de maquinari dependran del sistema operatiu sota el qual s'estigui executant l'aplicació. El requeriments de programari són els següents:

Entorn Windows:

- Windows 95 o superior.
- JRE 1.6 o superior.
- (opcional) Navegador Web i visualitzador de documents PDF.

Entorn Linux:

- Linux (kernel 2.6 o superior).
- JRE 1.6 o superior.
- (opcional) Navegador Web i visualitzador de documents PDF.

És recomanable també configurar la pantalla de la màquina on es vagi a executar l'aplicació a una resolució de 1024x768 punts o similar.

### 2.2 Instal·lació

El fitxer d'instal·lació de la plataforma PEX pot obtenir-se de dues maneres: mitjançant el CD adjunt a aquesta documentació o bé, via Internet, descarregant-lo des de la pàgina *web* de l'aplicació (disponible en breu).

L'arxiu s'anomena PEX.zip i haurà de ser emmagatzemat en algun lloc conegut del disc dur de l'ordinador en el que es desitgi instal·lar l'aplicació. Una vegada obtingut aquest arxiu, el sistema ja està preparat per dur a terme el procés d'instal·lació.

El contingut de l'arxiu PEX.zip és el següent:

Arxiu	Descripció
<b>\data</b>	Carpeta que conté, per defecte, informació referent als exàmens.
<b>\xsl</b>	Carpeta que conté les llibreries utilitzades per l'aplicació. En aquesta carpeta han de figurar els arxius avalon-framework-cvs-20020806.jar, batik.jar, fop.jar, xalan-2.4.1.jar, xercesImpl.jar i xml-apis.jar.
<b>\lib</b>	Carpeta que conté els fulls d'estil que utilitza la plataforma.
<b>\temp</b>	Carpeta temporal.
<b>PEX_Admin.jar</b>	És l'arxiu de configuració de l'aplicació.
<b>pex.cer</b>	Certificat digital que utilitza la plataforma.
<b>manual.pdf</b>	Aquest document.

El primer pas serà descomprimir l'arxiu PEX.zip a una carpeta, per exemple, a la carpeta 'pex'. Per a fer això

- amb Linux:
  - s'obre una finestra de terminal i es crea la carpeta *pex* amb la comanda 'mkdir pex'.
  - s'entra dins de la carpeta *pex* amb la comanda 'cd pex'.
  - s'executa l'aplicació unzip passant com a paràmetre l'arxiu PEX. 'unzip /ruta\_on\_hi\_hi\_l'arxiu/PEX.zip'
- amb Windows:
  - Si no està instal·lada l'aplicació Winzip, ens la descarreguem ([www.winzip.com](http://www.winzip.com)) i instal·lem.
  - Amb l'explorador de fitxers fem doble *clic* damunt l'arxiu PEX.zip.
  - S'executa Winzip. Es prem el boto 'Extract' de la barra d'eines i s'indica que es vol extreure l'arxiu a la carpeta *pex*.

Hi ha altres maneres de fer-ho tant amb Linux com amb Windows. Si tenim una de les darreres versions de Linux instal·lades, amb l'ajut de l'explorador d'arxius Nautilus, si fem *clic* damunt de l'arxiu PEX.zip s'executarà el descompressor d'arxius, es prem *Extract*, s'indica la carpeta on es vol descomprimir l'arxiu (/pex), es prem *Ok* i ja tenim l'arxiu descomprimit.



## **2.3 Accés a PEX**

Per tal d'accedir a l'aplicació d'administració, tant en entorn Windows com Linux, hi ha dues maneres:

1. Per línia de comandes, escrivint l'ordre:

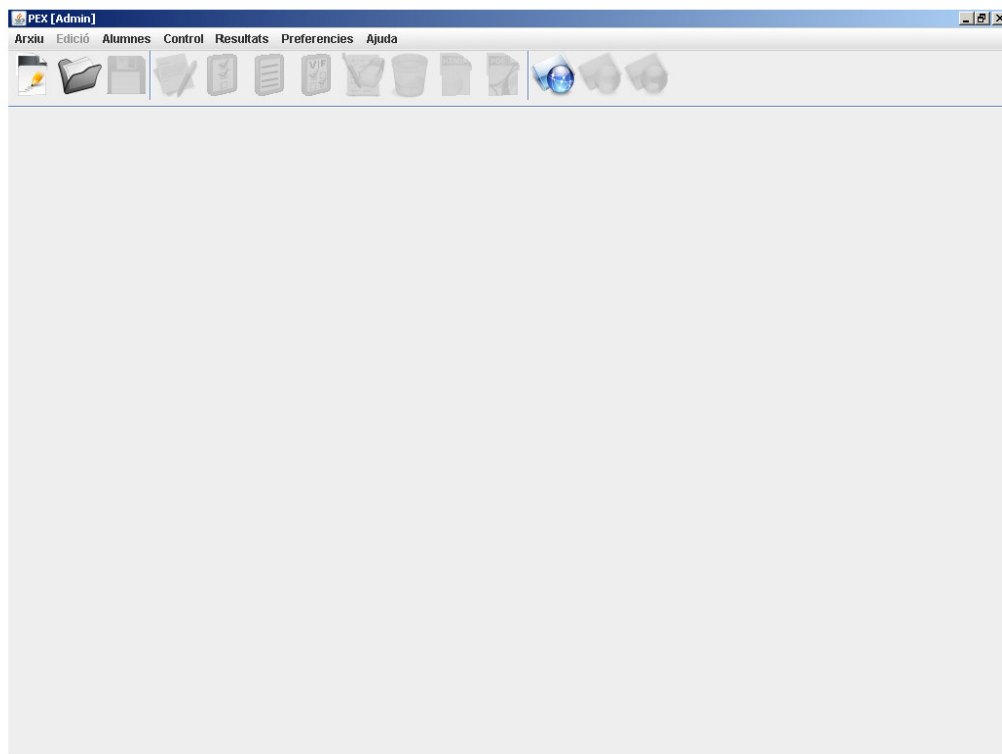
```
java -jar PEX__Admin_.jar
```

2. Pel navegador d'arxius del sistema operatiu, fent doble clic a la icona de l'aplicació PEX\_\_Admin\_.jar

## 3 Funcionament de PEX Admin

### 3.1 Elements de la pantalla inicial

Un cop s'ha accedit a l'aplicació, mitjançant alguna de les maneres indicades en el punt anterior, s'obre la pantalla inicial de PEX Admin recollida a la següent imatge.



Imatge 1 – Pantalla inicial

Des d'aquesta pàgina principal es poden fer les següents accions:

- **Crear o modificar un arxiu d'examen:**

Permet la creació d'un examen. Des d'aquesta opció es podran afegir, modificar i/o eliminar preguntes a un examen o modificar les propietats del mateix.

- **Crear o modificar un arxiu d'alumnes:**

Permet indicar quins alumnes podran realitzar un examen així com indicar per cada alumne una clau d'accés personal que permeti identificar-lo. Els llistat d'alumnes es pot importar o exportar, a més a més del format que utilitza la plataforma per guardar l'arxiu d'alumnes, a format CSV. El arxius en format CSV es poden utilitzar fàcilment en fulls de càlcul i bases de dades.

- **Desplegar un examen en xarxa:**

Permet la realització d'exàmens en xarxa. Es pot controlar l'inici, el final, la duració i qui pot realitzar l'examen. En tot moment es controla el progrés dels exàmens despleats com, per exemple, quins alumnes s'han connectat, quins han entregat l'examen...

- **Corregir un examen:**



Permet corregir un examen de forma automàtica. L'usuari escull quin tipus de preguntes vol corregir, les possibles penalitzacions per a determinats tipus de preguntes... Després de la correcció es poden generar informes amb els resultats obtinguts.

A més a més d'aquestes opcions principals, que vindrien a ser els quatre grans mòduls sobre els quals gira PEX, també es poden realitzar opcions d'impressió i exportació d'exàmens, ...

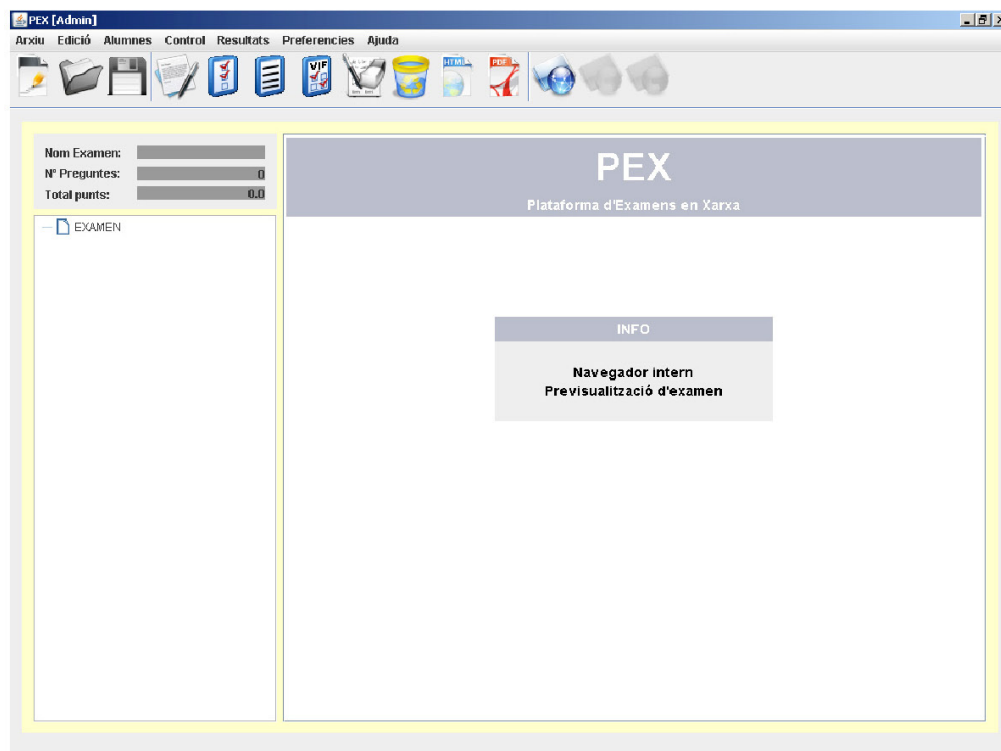
A continuació es descriuran amb més detall les opcions anteriors.

### 3.2 Crear o modificar un arxiu d'examen

Accés a la funció:

MENÚ	BARRA D'EINES
Arxiu -> Nou Examen	
Arxiu -> Obrir Examen	

La funcionalitat de crear un examen ens porta a la següent pantalla:



Imatge 2 - Crear examen


A la part esquerra de la interfície tenim l'arbre de preguntes i a la part dreta la zona de visualització de l'examen.

Des d'aquesta pantalla l'usuari pot:

- Introduir les propietats de l'examen.
- Afegir una pregunta
- Modificar una pregunta
- Eliminar una pregunta

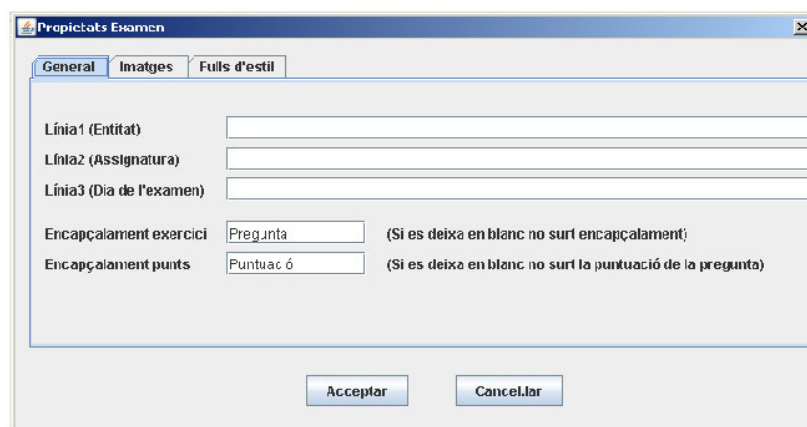
### 3.2.1 Propietats de l'examen

Accés a la funció:

MENÚ	BARRA D'EINES
Edició -> Propietats de l'examen	

Les propietats que es poden establir per un examen es divideixen en 3 blocs: propietats generals de l'examen, propietats de les imatges que utilitzem en l'examen i propietats de visualització de l'examen. Concretant aquestes propietats tenim:

- **General**



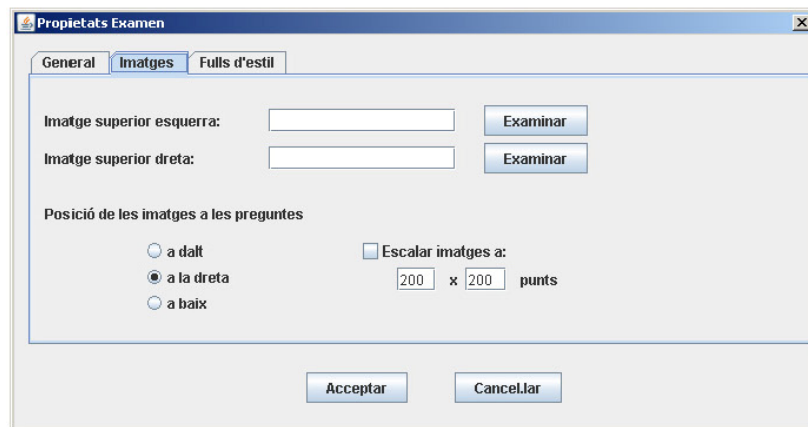
**Imatge 3 - Propietats generals**

Encapçalament d'examen. Es tracta de tres línies de text que l'usuari pot utilitzar com desitgi. S'aconsella utilitzar la primera línia per indicar el nom de la institució, la segona línia per indicar l'assignatura o matèria i la tercera per indicar la data de realització de l'examen.

La paraula identificadora d'encapçalament de pregunta. Per defecte serà 'Pregunta', però es pot canviar a Problema, Qüestió... Si es deixa en blanc només sortirà únicament el número de pregunta.

La paraula identificadora d'encapçalament de puntuació. Per defecte serà 'Puntuació'.

- **Imatges**



**Imatge 4 - Propietats imatges**

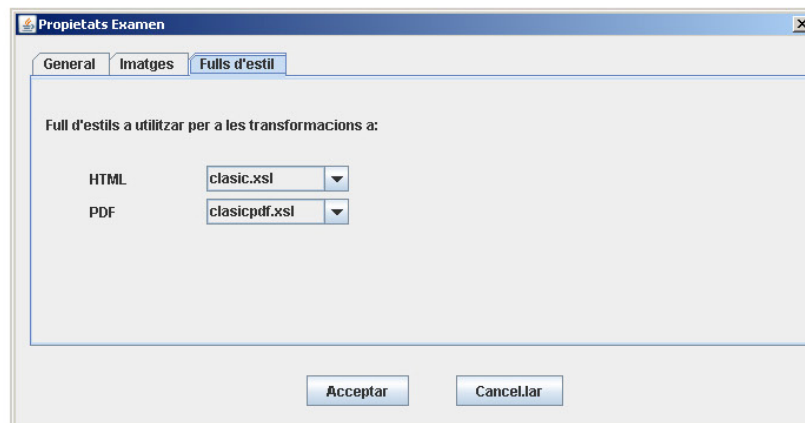
Establir una imatge a mostrar a la part superior esquerra de l'examen, com podria ser un logotip.

Establir una imatge a mostrar a la part superior dreta de l'examen.

Establir la posició de les imatges en una pregunta. La posició pot ser a dalt, a la dreta o a baix de la pregunta.

Indicar si es vol redimensionar la imatge o mostrar-la a tamany real. Les mesures indicades són en píxels.

- **Fulls d'estil**



**Imatge 5 - Propietats fulls d'estil**




Permet seleccionar quin full d'estils s'aplicarà a l'examen per a la seva visualització en format HTML i en format PDF. Aquests fulls d'estil es troben a la carpeta `xsl/html` i `xsl/pdf` per a les transformacions a format HTML i PDF respectivament. L'usuari pot afegir els seus propis fulls d'estil a la carpeta i posteriorment utilitzar-los.

Amb la plataforma, per defecte, es proporcionen 2 fulls d'estil per transformar l'examen, el full `clasic.xsl` (seleccionat per defecte) i `modern.xsl`.

Les propietats de l'examen es poden modificar en qualsevol moment, quedant reflectides les modificacions automàticament a l'instant.

### 3.2.2 Afegir pregunta

Accés a la funció:

MENÚ	BARRA D'EINES
Edició -> Afegir pregunta...	  

En la versió 1.0 de PEX s'ha contemplat la possibilitat d'afegir 3 tipus de preguntes:

**Preguntes de tipus test (PT1).** Preguntes en les quals s'ha d'escollir l'opció correcta. Poden ser de selecció simple o múltiple. Es possible especificar fins a 5 opcions. Poden estar subjectes a penalitzacions per error.

**Preguntes de contingut (PT2).** Preguntes en les quals s'ha de descriure alguna cosa. S'especifica la longitud en línies que es desitgi com a màxim per respondre la pregunta.

**Preguntes de veritat / fals (PT3).** Preguntes en les quals s'ha d'escollir si l'enunciat és veritat o és fals. Es poden especificar fins a 5 enunciats per pregunta. Estan subjectes a penalitzacions per error.

En cada pregunta s'especifica el número de pregunta que es desitja (automàticament es detecta però l'usuari el pot canviar) i la puntuació de la mateixa així com si es vol incloure-hi una imatge il·lustrativa.

Si s'utilitzen imatges en les preguntes, aquestes han d'estar guardades en la mateixa carpeta que l'arxiu de l'examen.

El quadre de diàleg per afegir una pregunta de tipus test seria el de la següent figura:


**Imatge 6 - Afegir pregunta tipus test (PT1)**

Cada pregunta té l'opció 'Afegir més' que de forma ràpida permet introduir preguntes del mateix tipus.



### 3.2.3 Modificar una pregunta

Accés a la funció:


MENÚ	MENÚ CONTEXTUAL	BARRA D'EINES
Edició -> Modificar pregunta	Botó dret del ratolí sobre la pregunta a modificar a l'arbre de l'examen -> Modificar pregunta	

Aquesta funcionalitat permet modificar qualsevol dels paràmetres de la pregunta seleccionada, ja sigui l'enunciat o les opcions de la mateixa.

La pregunta a modificar ha d'estar prèviament seleccionada a l'arbre de preguntes de l'examen.

### 3.2.4 Eliminar una pregunta

Accés a la funció:

MENÚ	MENÚ CONTEXTUAL	BARRA D'EINES
Edició -> Eliminar pregunta	Botó dret del ratolí sobre la pregunta a eliminar a l'arbre de l'examen -> Eliminar pregunta	

Aquesta funcionalitat elimina la pregunta seleccionada. Un cop realitzada aquesta opció, la pregunta no es podrà tornar a recuperar.

La pregunta a eliminar ha d'estar prèviament seleccionada a l'arbre de preguntes de l'examen.

### 3.3 Crear o modificar un arxiu d'alumnes

Accés a la funció:

MENÚ	BARRA D'EINES
Alumnes -> Editar	

La funcionalitat de crear un arxiu d'alumnes ens porta a la següent pantalla:

Imatge 7 - Edició d'alumnes

Des d'aquesta pantalla l'usuari pot:

- Introduir les dades dels alumnes
- Carregar un arxiu d'alumnes ja existent
- Guardar un arxiu d'alumnes
- Importar / Exportar el llistat d'alumnes a format CSV

Per tal d'introduir les dades d'un alumne l'únic que s'ha de fer es introduir les seves dades en una nova fila de la taula. Les dades que es requereixen de cada alumne són: cognoms, nom, DNI i clau d'accés per realitzar els exàmens. Es possible ordenar els les dades ascendentment o descendentment pel camp que es vulgui fent clic a la capçalera de la columna del camp que es desitja ordenar.

El DNI i la clau d'accés són els valors que haurà d'introduir l'alumne a l'hora de realitzar l'examen en xarxa per tal de verificar la seva identitat.


Un cop s'hagi escrit la clau, com a mesura de seguretat, aquesta serà automàticament xifrada utilitzant un hash amb SHA-1 de 160 bits. Per modificar la clau, s'ha d'eliminar el contingut de la cel·la i tornar a escriure la clau.

Es possible carregar un arxiu d'alumnes ja existent i modificar-ne les dades desitjades. En un arxiu d'alumnes també es pot especificar de quina assignatura i curs es tracta, tot i que no es requereix, es aconsellable fer-ho.

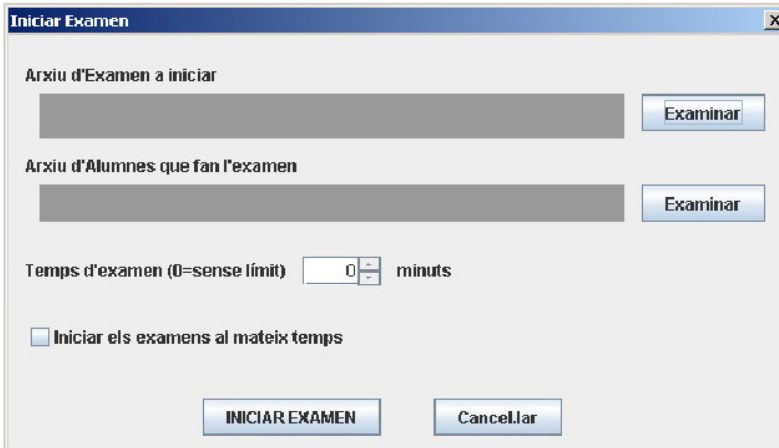
Es pot importar / exportar el llistat d'alumnes a format CSV per la seva posterior utilització en un programa extern (fulls de càlcul, bases de dades...).

### 3.4 Desplegar un examen en xarxa

Accés a la funció:

MENÚ	BARRA D'EINES
Control -> Iniciar Examen	

La funcionalitat de desplegar un examen en xarxa ens porta al següent quadre de diàleg:



**Iniciar Examen**

Arxiu d'Examen a iniciar

Arxiu d'Alumnes que fan l'examen

Temps d'examen (0=sense límit)    minuts

☐ Iniciar els examens al mateix temps

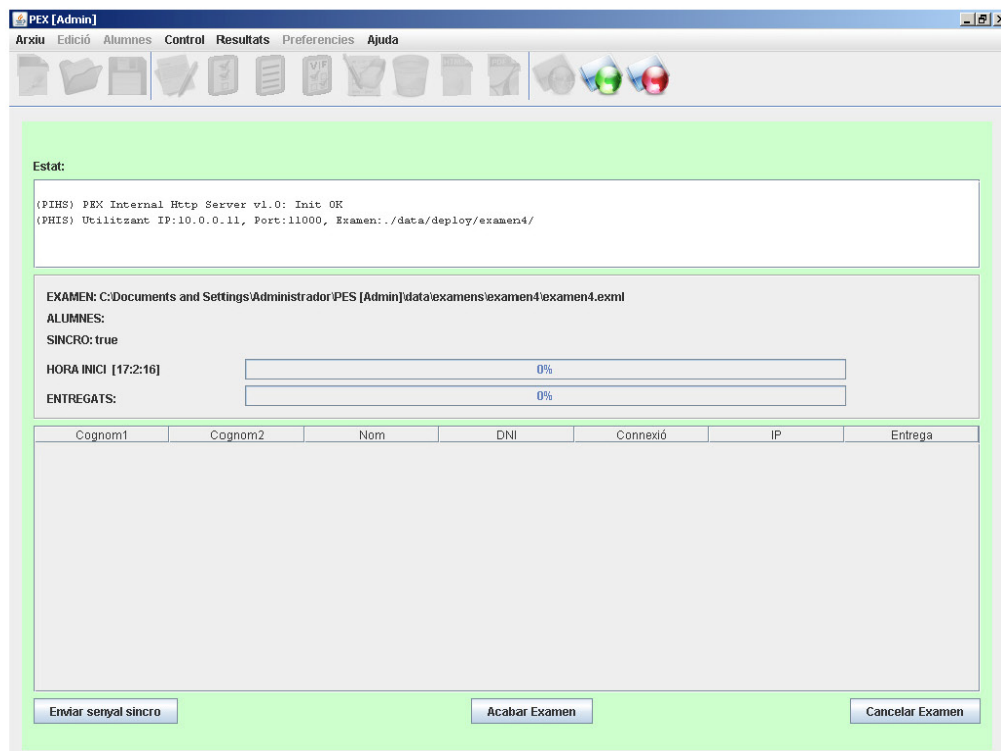
**Imatge 8 - Iniciar Examen**

En aquest quadre de diàleg s'han d'especificar les següents opcions:

1. L'examen que es vol desplegar en xarxa. S'ha d'especificar un arxiu d'examen generat prèviament pel mòdul de creació d'exàmens de PEX.
2. Els alumnes que poden fer l'examen. S'ha d'especificar un arxiu d'alumnes generat prèviament pel mòdul d'alumnes o bé, si es deixa en blanc, s'entendrà que qualsevol persona podrà realitzar l'examen desplegat accedint sense usuari ni clau.

3. El temps d'examen en minuts. Si el temps es posa a 0 minuts (així està per defecte), s'entén que no hi ha límit de temps. Quan el temps indicat s'esgoti, tots els exàmens desplegats s'entregaran automàticament. L'alumne veu en cada moment el temps que resta d'examen.
4. Si es desitja sincronitzar l'inici dels exàmens per a que aquest sigui el mateix per a tothom. Els alumnes es poden descarregar l'examen però no podran començar fins que s'envii la senyal d'inici.

Un cop verificades les dades proporcionades en el quadre de diàleg, es visualitza la següent pantalla de control d'examen:



**Imatge 9 - Examen en xarxa iniciat**

Des d'aquesta pantalla l'administrador pot visualitzar l'estat dels exàmens desplegats i altres paràmetres en temps real, com el temps que falta per finalitzar l'examen (si així s'ha especificat), quins alumnes s'han connectat, des de quina direcció IP, i si s'han descarregat i finalitzat l'examen.

Les accions bàsiques que es poden fer en aquesta pantalla són tres:

1. Enviar la senyal d'inici d'examen. Si s'ha especificat que es vol sincronitzar l'inici dels exàmens, aquests no es visualitzaran i per tant no començaran fins que s'envii la senyal de sincronització d'inici.
2. Acabar l'examen. Fent clic en aquesta opció es dona per acabat un examen desplegat en xarxa. Es guarden les dades i els resultats dels exàmens.
3. Cancel·lar examen. Es cancel·la el desplegament en xarxa d'un examen. No es guarda cap dada ni resultat.

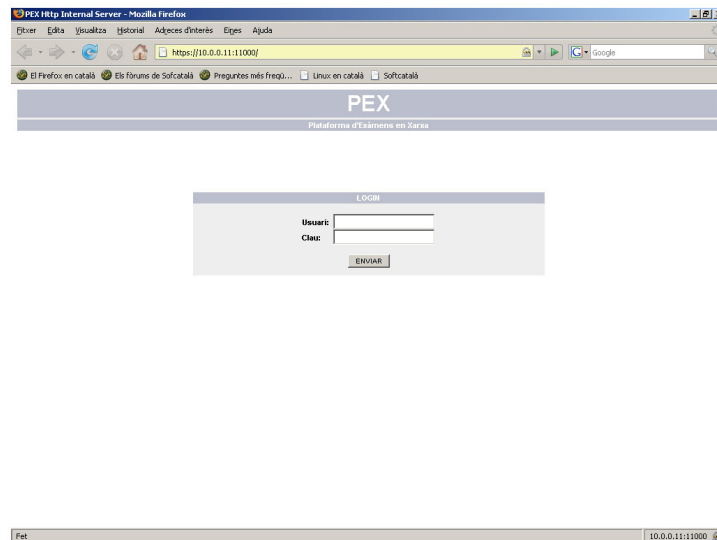
### 3.4.1 PEX Client

Un cop arribats a aquest punt, els alumnes ja es poden connectar a PEX Admin per poder descarregar l'examen que han de realitzar. L'únic que han de fer es obrir al seu ordinador un navegador web i introduir la direcció formada per la direcció IP i el port que es visualitzen a la finestra d'Estat i que prèviament han estat detectats per PEX. Per exemple, en el cas de desplegament de la imatge 9, la direcció URL a introduir seria: <https://10.0.0.11:11000>. L'accés es realitza mitjançant el protocol HTTPS per motius de seguretat.

Si no apareix cap direcció IP a la finestra d'estat, significa que no s'ha pogut detectar automàticament i per tant, aquesta s'haurà de configurar manualment a Preferències -> Opcions.

Un cop introduïda la direcció URL pertinent, i després d'acceptar els certificats de seguretat, apareixerà la pantalla d'accés de PEX Admin.

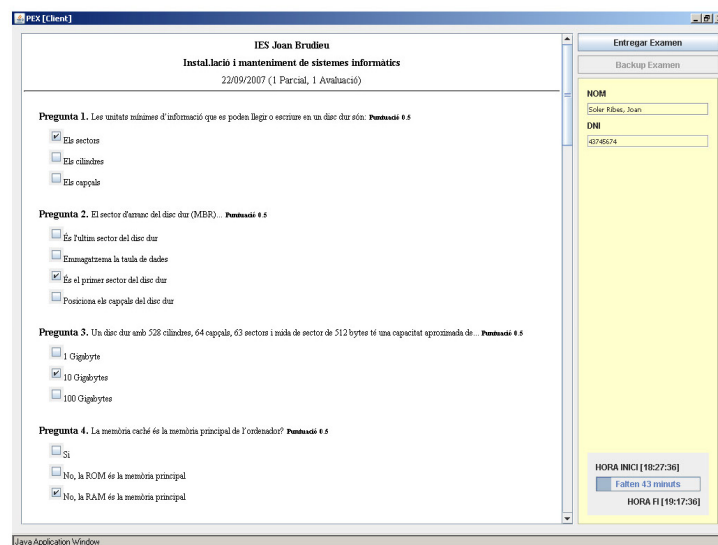
En el formulari d'accés, els alumnes hauran d'introduir les seves dades (DNI i clau) especificades en el fitxer d'alumnes relacionat amb l'examen desplegat. Si la identificació es incorrecta es mostrarà l'avís pertinent. Un cop identificats correctament, es descarregarà i s'executarà automàticament PEX Client.



Imatge 10 - Pantalla d'accés web

PEX Client és l'eina de la plataforma que s'utilitza per a que un alumne faci l'examen. La seva estructura es molt simple. A la part esquerra es mostra l'examen i a la dreta les dades de l'alumne juntament amb el temps restant d'exàmens (en el cas de que es tracti d'un examen amb temps limitat). Un cop l'alumne hagi realitzat l'examen pot prémer el botó 'Entregar examen' per finalitzar l'examen. Les seves respostes seran enviades a l'aplicació d'administració PEX Admin.

Una mostra d'aquesta aplicació es pot veure en la següent imatge:



Imatge 11 - PEX Client

### 3.5 Corregir un examen

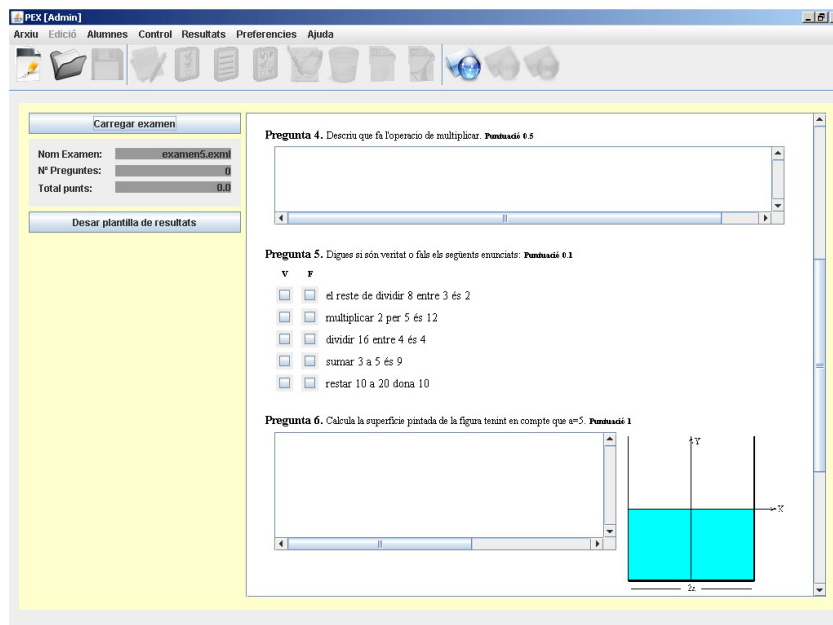
Per corregir un examen que ja ha estat desplegat en xarxa s'han de seguir, per ordre, els 3 passos següents:

#### 1. Crear una plantilla de correcció

Accés a la funció:

MENÚ	BARRA D'EINES
Resultats -> Crear plantilla de correcció	

La funcionalitat de crear una plantilla de correcció ens porta a la següent pantalla:



Imatge 12 - Crear plantilla de correcció

Els passos a seguir per crear una plantilla de correcció són els següents:

- Carregar l'examen del qual volem crear la plantilla de correcció.
- Respondre a les preguntes de l'examen.
- Un cop realitzat l'examen, en guardem el resultat.

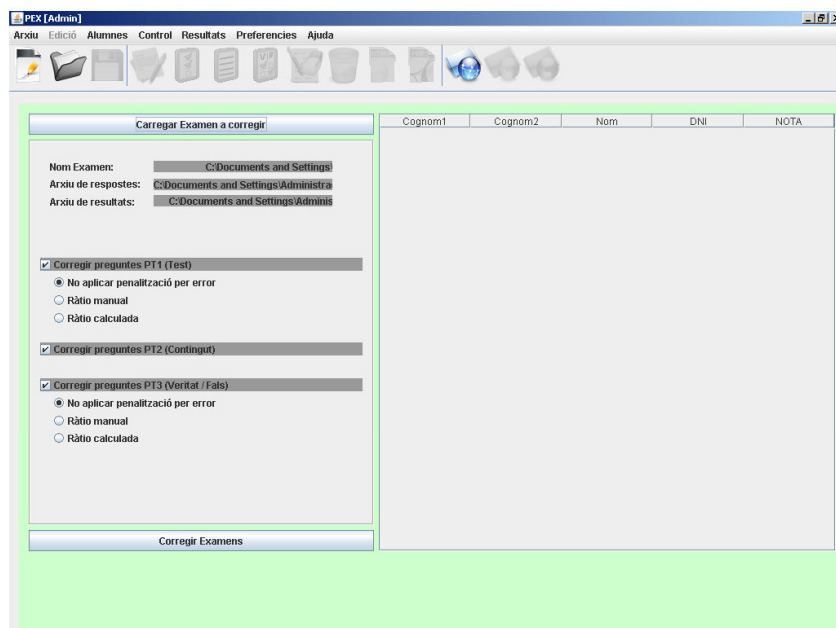


## 2. Corregir l'examen

Accés a la funció:

MENÚ	BARRA D'EINES
Resultats -> Corregir examen	

La funcionalitat de corregir un examen ens porta a la següent pantalla:



Imatge 13 - Corregir examen

Els passos a seguir per corregir un examen són els següents:

- Carregar l'examen a corregir.
- Seleccionar quin tipus de preguntes volem corregir i si es el cas, quin tipus de penalització per error s'haurà d'aplicar (manual o calculada).
- Fer clic a corregir examen.

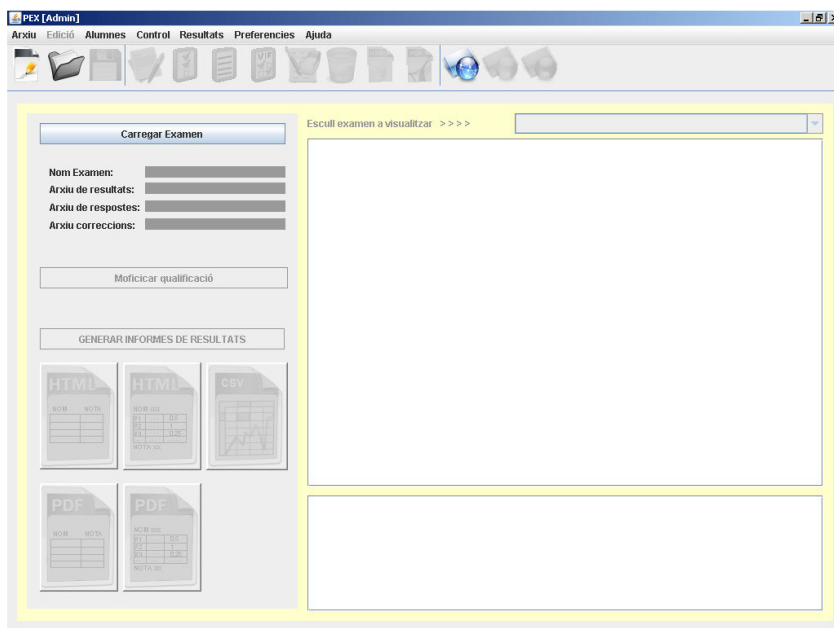
Veurem com les qualificacions obtingudes per tots els alumnes que han fet l'examen van apareixent a la taula de la dreta.

### 3. Visualitzar els resultats

Accés a la funció:

MENÚ	BARRA D'EINES
Resultats -> Visualitzar resultats	

La funcionalitat de visualitzar resultats ens porta a la següent pantalla:



**Imatge 14 - Visualització de resultats**

Els passos a seguir per visualitzar els resultats d'un examen són els següents:

- Carregar l'examen del qual volem veure els resultats.
- Prémer el botó de generar resultats. Aquesta acció genera 2 informes, un de simple, amb les dades de l'alumne i la nota final que ha obtingut i un altre de detallat amb tot el procés de correcció. Aquest 2 informes es generen en format HTML i PDF.
- Navegar per les diferents opcions per visualitzar els resultats. Es poden visualitzar els informes generats, l'examen que ha fet un determinat alumne o afegir o canviar la puntuació obtinguda d'alguna pregunta d'algun alumne.

### 3.6 Darreres consideracions

Les accions de crear una plantilla de correcció, corregir un examen i visualitzar resultats generen una sèrie de carpetes a dins de la carpeta de l'examen que s'està tractant. Aquestes carpetes no s'han d'eliminar doncs contenen informació important.

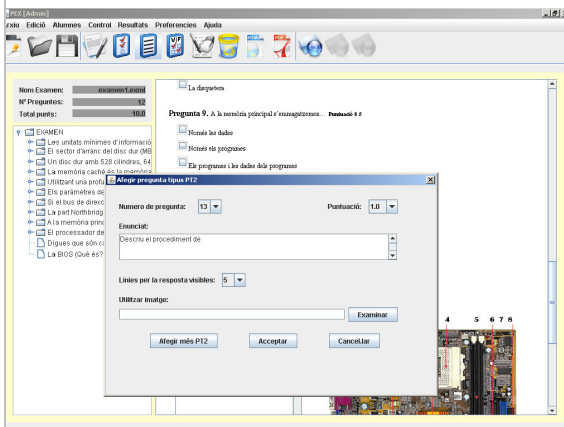
La ràtio calculada considera automàticament la penalització que s'ha d'aplicar a una pregunta en funció de que l'esperança de la nota d'un alumne que contesti a l'atzar sigui 0.

A Preferències -> Opcions podem establir:

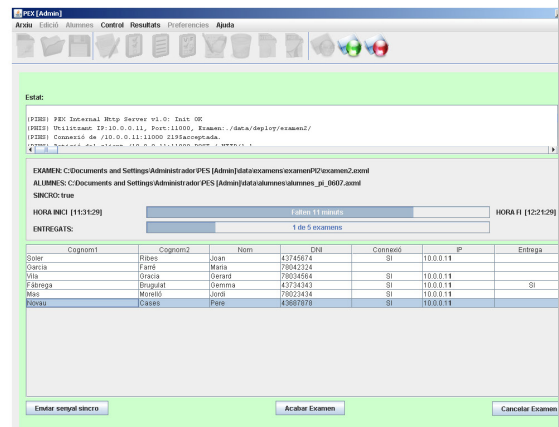
- La IP i el port que utilitza PHIS (PEX HTTP Internal Server) per escoltar i rebre peticions al desplegar un examen.
- El port que utilitza PEX Client per comunicar-se amb PHIS.
- A l'eliminar una pregunta es re indexa automàticament el número de pregunta si aquesta opció està seleccionada.
- Quan utilitzem una imatge en una pregunta, aquesta a d'estar copiada a la carpeta de l'examen que estem treballant. Si es selecciona aquesta opció, la imatge es copia automàticament a la carpeta de l'examen si aquesta no i era.
- La ràtio manual a descomptar en cas d'error a una pregunta amb penalització per error. Aquesta ràtio pot ser un descompte de X punts per cada error o en funció del nombre de preguntes errades.
- El caràcter delimitador en una exportació a CSV, doncs alguns programes utilitzen el caràcter ';' (Microsoft Excel) com a delimitador de camps i altres (Open Office Calc) el caràcter ','.

## 5 Annexes

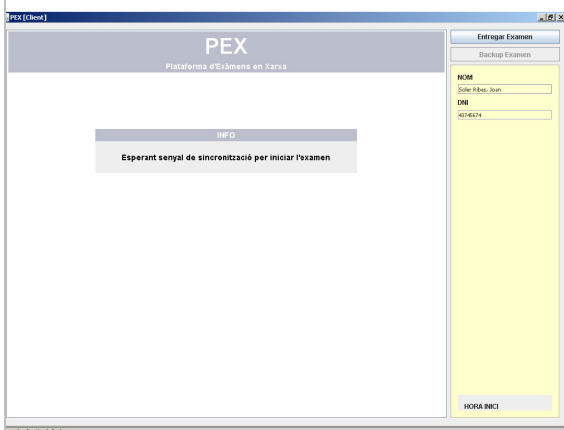
### 5.1 Captures de l'aplicació en funcionament



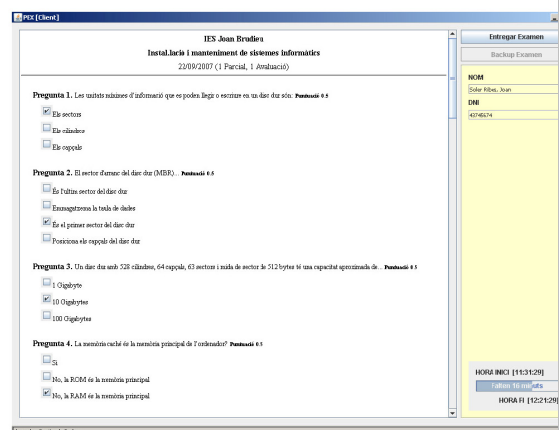
Imatge 15 - Procés de creació d'examen



Imatge 16 - Examen desplegat en xarxa



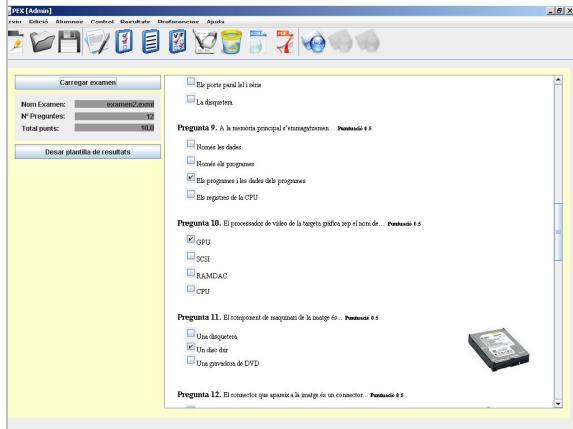
Imatge 17 - Client esperant senyal d'inici



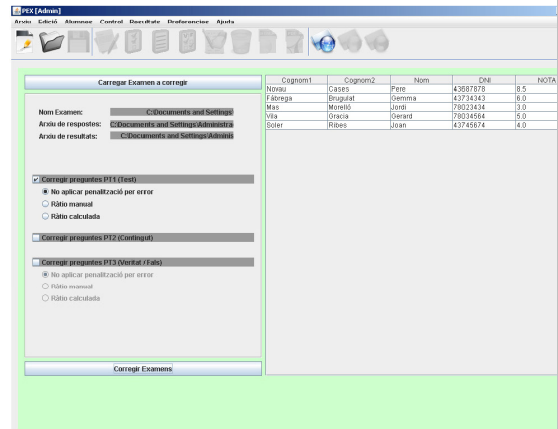
Imatge 18 - Client realitzant l'examen

## Manual d'usuari

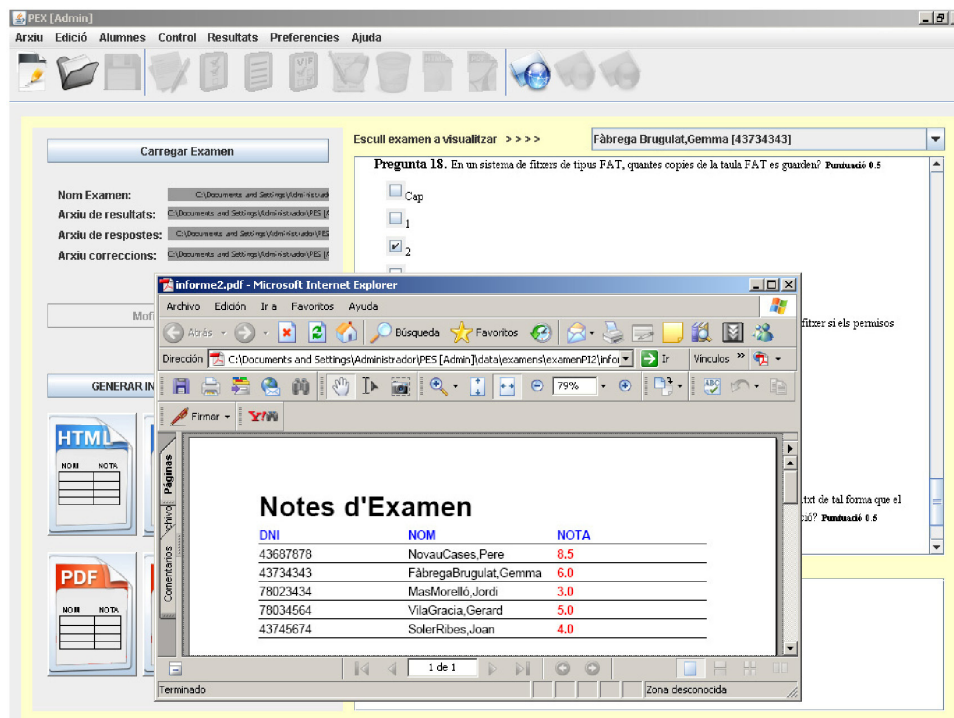
PEX 1.0



Imatge 19 - Plantilla de correcció



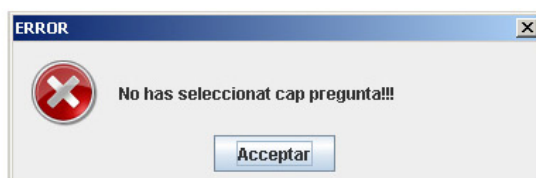
Imatge 20 - Corregint l'examen



Imatge 21 - Visualitzant els resultats

## 5.2 Llistat dels errors més freqüents

Quan es produeix un error, es mostra un quadre de diàleg, com el de la imatge següent, tot indicant la causa de l'error.



Imatge 22 - Missatge d'error

Els errors més freqüents que es produeixen són els següents:

CODI	ERROR	DESCRIPCIÓ
1	Examen no guardat!!!	Per realitzar determinades accions, com pot ser una previsualització, prèviament s'ha d'haver guardat l'examen.
2	No hi ha cap examen obert!!!	Per realitzar determinades accions, com imprimir, prèviament s'ha d'haver obert o creat un examen.
3	Número de pregunta repetit!!!	S'està intentant afegir una pregunta amb un número de pregunta existent a l'examen.
4	Arxiu d'alumnes incorrecte!!!	Al carregar, guardar, importar o exportar un arxiu d'alumnes el seu contingut no sembla correcta.
5	No has seleccionat cap pregunta!!!	Per realitzar accions sobre una pregunta, com modificar o eliminar, s'ha d'haver seleccionat prèviament.
6	Arxiu de resultats i/o respostes no trobats!!!	Per realitzar una correcció d'examen s'han de generar els arxius per ordre, tal i com s'indica en el present manual. En cas contrari es molt probable que faltin arxius.

### 5.3 Taula d'il·lustracions

Imatge 1 – Pantalla inicial .....	7
Imatge 2 - Crear examen .....	9
Imatge 3 - Propietats generals .....	10
Imatge 4 - Propietats imatges .....	11
Imatge 5 - Propietats fulls d'estil .....	11
Imatge 6 - Afegir pregunta tipus test (PT1).....	13
Imatge 7 - Edició d'alumnes .....	15
Imatge 8 - Iniciar Examen .....	17
Imatge 9 - Examen en xarxa iniciat.....	18
Imatge 10 - Pantalla d'accés web.....	20
Imatge 11 - PEX Client .....	20
Imatge 12 - Crear plantilla de correcció .....	21
Imatge 13 - Corregir examen.....	22
Imatge 14 - Visualització de resultats.....	23
Imatge 15 - Procés de creació d'examen.....	25
Imatge 16 - Examen desplegat en xarxa.....	25
Imatge 17 - Client esperant senyal d'inici.....	25
Imatge 18 - Client realitzant l'examen.....	25
Imatge 19 - Plantilla de correcció.....	26
Imatge 20 - Corregint l'examen .....	26
Imatge 21 - Visualitzant els resultats.....	26
Imatge 22 - Missatge d'error.....	27

## Capítol 5

# Conclusions i futures línies de treball

### 5.1 Conclusions

Tot i que en un principi poguéssim semblar una idea complexa i difícil, crec que s'han complert els requeriments inicials establerts i el resultat ha estat més que satisfactori.

Amb el disseny modular, cada part interacciona perfectament amb l'altra, formant d'aquesta manera una aplicació que es pot desglossar en mòduls pràcticament independents.

Un fet a tenir molt en compte es gairebé la nul·la instal·lació que requereix la plataforma i la seva facilitat d'ús. Es pot portar instal·lada en qualsevol suport com per exemple, en una memòria USB. Aquesta memòria es connecta a l'ordinador que ha de fer les funcions d'administrador i la plataforma ja està llesta per funcionar.



Amb aquesta facilitat d'instal·lació, s'eviten les molestes accions que s'han de realitzar en moltes aules d'informàtica amb determinat tipus de programari instal·lat que protegeixen la màquina de canvis de programari.

També considerar que tot el programari utilitzat és programari lliure i les diferents tecnologies són multi plataforma, d'aquesta manera no es limita l'aplicació en cap sentit.

Definitivament una eina molt útil per als qui en el seu dia a dia hagin de realitzar aquest tipus de proves.

Les conclusions personals que extrec respecte cada una de les tecnologies i programari emprat és el següent:

### **Respecta Java**

Un llenguatge de programació que ofereix totes les funcionalitats que un programador pot desitjar. La documentació i el suport a l'usuari són impressionants.

### **Respecte XML**

Proporciona molta flexibilitat per al tractament de les dades. Sens dubte un estàndard a seguir. És fàcil d'utilitzar i amb els APIs de Java SAX i DOM, el tractament de XML resulta molt simple.

### **Respecte XSL**

És fàcil d'utilitzar pel que fa a les transformacions que empren XSLT, però quan s'ha d'utilitzar XSL-FO per a realitzar les transformacions la cosa es complica bastant més. Aquí ja entren en joc més factors, com per exemple el motor de renderització que produirà la sortida i quins FO té implementats aquest.

### **Respecte FOP**

Bastant difícil de treballar ja que està limitat pels FO que té implementats i que no són ni de bon tros tots. Acceptable si no és vol formatar taules o alguna estructura de presentació complexa. A part, la informació que dona quan produeix un error és confusa i poc rellevant sobre la causa que l'ha provocat.

## **Respecte SSL**

Un protocol molt complet i útil si es busca seguretat. Potser resulta una mica complicat treballar amb ell degut a la seva complexitat.

## **5.2 Futures línies de treball**

Són moltes les futures línies de treball a desenvolupar a la plataforma. D'entre aquestes caldria destacar les següents:

- Incorporar nous tipus de preguntes. Actualment la plataforma suporta 3 tipus diferents de preguntes en un examen: tipus test, contingut i veritat o fals. Tot i que donen força joc, es podrien incorporar nous tipus de preguntes com, emplenar espais en blanc, relacionar conceptes...
- Incorporar un corrector ortogràfic. Actualment la plataforma no disposa de corrector ortogràfic. Sembla que incorporar un corrector ortogràfic com ISPELL no ha de resultar molt complicat.
- Internacionalitzar adequadament l'aplicació seria un avenç important, és a dir, introduir la possibilitat d'afegir idiomes de cara a presentar la interfície i els diferents missatges que pugui donar l'aplicació en l'idioma seleccionat per l'usuari i d'aquesta manera fer l'aplicació accessible a més usuaris.

En principi, aquests canvis es duran a terme en un període curt de temps ja que crec que amb aquests canvis pot resultar una plataforma molt completa i molt útil per a administrar tot el procés que involucra la realització d'un examen.

# Taula d'il·lustracions

Il·lustració 1. Exemple arxiu XML.....	9
Il·lustració 2. Exemple arxiu XSL .....	12
Il·lustració 3. Exemple arxiu jnlp .....	20
Il·lustració 4. SSL handshake .....	22
Il·lustració 5. Temporització del projecte.....	26
Il·lustració 6. Taula de característiques .....	28
Il·lustració 7. Arquitectura de l'aplicació.....	29
Il·lustració 8. Cicle vital d'un examen .....	30
Il·lustració 9. Diagrama de casos d'ús .....	30
Il·lustració 10. Estructura d'un document d'examen.....	31
Il·lustració 11. Estructura d'un document d'alumnes .....	32
Il·lustració 12. Procés de desplegament en xarxa.....	34
Il·lustració 13. Certificat SSL .....	35
Il·lustració 14. Arxiu jnlp de desplegament.....	35
Il·lustració 15. Procés de correcció.....	36

# Glossari de termes

<b>API</b>	<i>(Aplication <b>P</b>rogram <b>I</b>nterface)</i> Conjunt de procediments i mètodes que es poden utilitzar en un programa per a la realització d'un conjunt de tasques específiques que proporciona l'API en qüestió.
<b>AWT</b>	<i>(Abstract <b>W</b>indowing <b>T</b>oolkit)</i> Col·lecció de mòduls d'interfícies gràfiques proveïdes per Sun dins del conjunt d'eines de desenvolupament Java.
<b>CERN</b>	<i>(<b>C</b>onseil <b>E</b>uropeen pour la <b>R</b>echerche <b>N</b>ucleaire)</i> Institució europea, creada el 1954, amb seu a Ginebra, que va desenvolupar per a les seves necessitats internes el primer navegador i el primer servidor WWW. Ha contribuït i segueix contribuint a la difusió d'aquesta tecnologia. És un dels rectors del W3C, organisme clau per l'estandarització de la WWW.
<b>direcció IP</b>	Grup de quatre bytes que identifiquen les màquines de la xarxa sota el protocol IP. Els bytes es representen, normalment, en format decimal, de 0 a 255. Per exemple. 192.24.84.1
<b>DOM</b>	<i>(<b>D</b>ocument <b>O</b>bject <b>M</b>odel)</i> DOM és un analitzador de documents XML. Representa els elements i continguts del document com una sèrie d'objectes disposats en un arbre jeràrquic que l'aplicació o llenguatge de programació pot manipular fàcilment.
<b>DTD</b>	Especificació per a un document SGML,HTML o XML que determina els elements estructurals i definicions de marca que podran utilitzar-se per a crear document d'aquests tipus. Pot ser interna o externa depenent d'on s'emmagatzema l'especificació, si al mateix document que conté la informació o en un document extern.
<b>hiperenllaç</b>	Punter existent en un document d'hipertext que enllaça a un altre document que pot ser o no de la mateixa modalitat.

<b>hipertext</b>	Modalitat de document en el que algunes paraules denominades hiperenllaços, o simplement enllaços, permeten accedir a altres parts del document, o el que és més freqüent en Internet, a altres documents.
<b>HTML</b>	<i>(HyperText Markup Language)</i> Llenguatge SGML considerat com el llenguatge estàndard per a la publicació de documents web. Admet components d'hipertext i multimèdia.
<b>HTTP</b>	<i>(HyperText Transmission Protocol)</i> Protocol utilitzat per la transferència de documents WWW.
<b>Internet</b>	Conjunt de xarxes interconnectades entre si que permeten la comunicació entre elles mitjançant la utilització del protocol TCP/IP.
<b>J2SE</b>	<i>(Java 2 Standard Edition)</i> Conjunt format pel llenguatge Java, la màquina virtual Java i els pertinents APIs.
<b>Java</b>	Llenguatge de programació orientat a objectes dissenyat expressament per a ser distribuït en un entorn de xarxa, Internet principalment, i ser independent de la plataforma.
<b>JDK</b>	<i>(Java Development Kit)</i> Conjunt de components necessaris per a desenvolupar en Java, per exemple, el J2SE.
<b>JNDI</b>	<i>(Java Naming and Directori Interface)</i> És un API que proveeix funcionalitats de nom i directori a aplicacions escrites en Java.
<b>JRE</b>	<i>(Java Runtime Environment)</i> Programari que necessita un ordinador per poder executar qualsevol programa Java.
<b>JVM</b>	<i>(Java Virtual Machine)</i> Intèrpret de Java que executa el codi de bytes en una plataforma particular.
<b>Linux</b>	Sistema Operatiu emmarcat dins del moviment del software lliure. Pot utilitzar-se en moltes plataformes informàtiques tot i que essencialment en les de tecnologia PC.

El seu creador va ser Linus Torvals de la Universitat de Helsinki (Finlàndia) que en 1991 va voler crear un sistema similar a UNIX però compatible amb tots els models de PC basats en Intel.

Linux, sense deixar de ser software lliure, es pot presentar en versions comercials com poden ser les distribucions Debian, Red Hat o S.u.S.e. per citar-ne algunes de les comuns i utilitzades.

**namespaces** Convenció avançada de XML. S'utilitzen instruccions de procés per assignar identificadors a URIs, un mecanisme amb el qual s'evita que els noms en un programa puguin ser amb confosos amb noms referents a un altre programa.

**navegador** (*browser*) Aplicació que serveix per a visualitzar documents WWW i navegar per Internet. En la seva forma més bàsica són aplicacions hipertext que faciliten la navegació pels servidors d'informació d'Internet; els més avançats compten amb funcionalitats multimèdia i permeten indistintament la navegació per servidors WWW, FTP, Gopher, news, e-mail...

**pàgina Web** Document escrit en qualsevol dels llenguatges (HTML, VRML, XML,...), que són interpretats pels navegadors.  
La pàgina pot estar tant a Internet, com dins d'una intranet, o simplement emmagatzemada en qualsevol dispositiu als que tingui accés l'ordinador.

**SAX** (*Simple Api for XML*)  
SAX és un analitzador de documents XML. La seva característica principal és que està basat en events.

**SGML** (*Standard Generalized Markup Language*)  
Estàndard per a llenguatges d'edició, particularment per a aquells que s'utilitzen per al intercanvi, maneig i publicació electrònica de documents.

**Sistema Operatiu** (SO)  
Conjunt de programes que controlen els recursos del sistema i serveixen d'intermediaris entre les aplicacions i els mateixos recursos. Windows, UNIX, Linux... són exemples de sistemes operatius.

<b>software lliure</b>	Software que pot ser copiat, distribuït i modificat lliurement. Aquest fet porta implícit la disponibilitat del codi font, però no implica en cap cas que sigui gratuït.
<b>SSL</b>	<i>(Secure Socket Layer)</i> Protocol que utilitza tècniques criptogràfiques per tal de garantir la seguretat i proporciona comunicacions segures a través de la xarxa.
<b>subclasse</b>	(Java) Classe que ha rebut certes característiques, mitjançant herència, d'una altra que passa a denominar-se superclasse.
<b>superclasse</b>	(Java) Classe que ha transmès a una altra (subclasse), mitjançant herència, un conjunt de característiques.
<b>URI</b>	<i>(Uniform Resource Identifier)</i> Identificador Uniforme de Recursos. Joc més general de noms i direccions que utilitzen cadenes separades per barres ( / ) per a referir-se als recursos.
<b>URL</b>	<i>(Uniform Resource Locator)</i> Sistema unificat d'identificació de recursos a la xarxa. Actualment i segons especificació del W3C, que prefereix URI, ja no s'ha d'utilitzar en els documents tècnics. Les URL es componen d'una part que fa referència al protocol utilitzat: HTTP, FTP..., etc; el nom del domini complet del servidor i la localització de la pàgina dins del servidor.
<b>W3C</b>	<i>(World Wide Web Consortium)</i> Organització apadrinada pel MIT i el CERN, entre altres, amb l'objectiu principal d'establir els estàndards relacionats amb la WWW.
<b>Web</b>	S'utilitza com a abreviació de WWW.
<b>WWW</b>	<i>(World Wide Web)</i> Sistema d'informació distribuït, amb mecanismes d'hipertext creat pels investigadors del CERN a Suïssa.
<b>XHTML</b>	<i>(eXtended HTML)</i> HTML estès. Reformulació del HTML 4.0 basada en una aplicació del XML 1.0

- XML** (*eXtensive Markup Language*)  
Subconjunt del SGML desenvolupat amb la finalitat de fer més compatible el propi SGML amb la xarxa.
- XSL** (*eXtensible Style Language*)  
Mecanisme avançat de fulls d'estil que ofereix als navegadors o a les aplicacions que ho requereixen informació de presentació i format.
- XSLT** (*XSL Transformations*)  
És la part de XSL que s'utilitza per a transformar un document XML en un altre document XML, o algun altre tipus de document que sigui reconegut per un navegador, com poden ser HTML i XHTML
- XSL-FO** (*XSL Formatting Objects*)  
Implementació específica d'un determinat XSLT, que aplicats a un fitxer XML en proporcionen un altre amb la informació del primer renderitzada a algun format de visualització impresa.



---

# Bibliografia

## Llibres:

- [1] S. Oaks. *Java Security*. O'REILLY, Canadà, 2002.
- [2] B. McLaughlin. *Java & XML*. O'REILLY, Canadà, 2001.
- [3] M. Morgan. *Descubre Java 1.2*. Prentice Hall, Madrid, 1999.
- [4] N. Pitts. *XML*. Anaya Multimedia, Madrid, 1999.

## Webs:

- [5] Forum Java de Sun, <http://forum.java.sun.com/index.jspa>
- [6] Tutorial de XSL, <http://geneura.ugr.es/~jmerelo/XSLT/>
- [7] XSLT Reference, <http://www.zvon.org/xxl/XSLTreference/Output/index.html>
- [8] Apache FOP, <http://xmlgraphics.apache.org/fop/>
- [9] Tutorial de FO, <http://geneura.ugr.es/~victor/cursillos/xml/FO/>
- [10] Tutorial de FO, <http://www.ecrion.com/Support/Resources/XSL-FOTutorial/Toc.xml.html>
- [11] Java en castellano, <http://www.programacion.com/java/tutorial>
- [12] Programación en castellano (XML), <http://www.programacion.net/articulos/XML/>
- [13] TCEXAM, <http://tcexam.tecnick.com>
- [14] Moodle, <http://moodle.org>

## Especificacions:

### [15] HTML40

W3C World Wide Web Consortium. *Especificació HTML 4.0*. Recomanació del W3C. Veure <http://www.w3.org/TR/REC-html40>

### [16] XML10

W3C World Wide Web Consortium. *Extensible Markup Language (XML) 1.0*. Recomanació del W3C. Veure <http://www.w3.org/TR/1998/REC-xml-19980210>

---

**[17] XSL10**

W3C World Wide Web Consortium. *Extensible Stylesheet Language (XSL) 1.0*.  
Recomanació del W3C. Veure <http://www.w3.org/TR/xsl>

**[18] XSLT10**

W3C World Wide Web Consortium. *XSL Transformations (XSLT) 1.0*.  
Recomanació del W3C. Veure <http://www.w3.org/TR/xslt>

**[19] Java™ 2 Platform, Standard Edition, v 1.6 Documentation**

Sun Microsystems. Veure <http://java.sun.com/javase/6/docs>